National Research Council Canada
Conseil national de recherches Canada

Division of Electrical Engineering
Division de génie électrique

ERB-994

Dynamic Programming in Pattern Recognition

H. Yamada and T. Kasvand

February 1987

87-04600/

Copyright 1987 by
National Research Council of Canada

Permission is granted to quote short excerpts and to reproduce figures and tables from this report, provided that the source of such material is fully acknowledged.

Copyright 1987 par
Conseil national de recherches du Canada

Il est permis de citer de courts extraits et de reproduire des figures ou tableaux du présent rapport, à condition d'en identifier clairement la source.

Additional copies are available free of charge from:

Editorial Office, Room 301
Division of Electrical Engineering
National Research Council of Canada
Ottawa, Ontario, Canada
K1A 0R6

Des exemplaires supplémentaires gratuits peuvent être obtenus à l'adresse suivante:

Bureau des publications, Pièce 301
Division de génie électrique
Conseil national de recherches du Canada
Ottawa (Ontario) Canada
K1A 0R6

DYNAMIC PROGRAMMING IN PATTERN RECOGNITION

H. Yamada and T. Kasvand

ERB-994

NRCC No. 27480

February 1987

TABLE OF CONTENTS

		Page
ABSTRACT		vii
RÉSUMÉ		viii
1	OVERVIEW	1
2	DYNAMIC PROGRAMMING IN IMAGE PROCESSING AND PATTERN RECOGNITION	3
	2.1 Formulation of a Matching Problem	3
	2.2 Introduction of Dynamic Programming	5
	2.3 Monotonic Continuous Dynamic Programming in Image Processing and Pattern Recognition	7
	2.4 Two-dimensional Neighbor Dynamic Programming Without Model	9
	2.5 Two-dimensional Inductive Dynamic Programming on Input Pixel	10
	2.6 Locally Ordered Dynamic Programming on Feature Element	11
	2.7 Further Expansions	13
3	RECOGNITION OF KIDNEY GLOMERULUS BY INDUCTIVE DYNAMIC PROGRAMMING MATCHING	18
	3.1 Introduction	18
	3.2 Inductive Dynamic Programming Matching Method	18
	3.3 Expansions of the Dynamic Programming	22
	3.4 Experiments	25
	3.5 Conclusions	27
4	RECOGNITION OF OCCLUDED, REFLECTIVE, AND TRANSPARENT OBJECTS WITH UNCONSTRAINED BACKGROUND AND ILLUMINATION BY USING INDUCTIVE DYNAMIC PROGRAMMING MATCHING	30
	4.1 Introduction	30
	4.2 Two-dimensional Dynamic Programming Matching Method	31
	4.3 Experiments with Scissors and Metal Rings	33
	4.4 Experiments with Transparent Ashtrays	38
	4.5 Discussion	40
5	ROTATION-INVARIANT CONTOUR DYNAMIC PROGRAMMING MATCHING METHOD FOR THREE-DIMENSIONAL OBJECT RECOGNITION	42
	5.1 Introduction	42
	5.2 Input and Model Representation and Global Distance	43
	5.3 Rotation-free Dynamic Programming Matching Method	47
	5.4 Conclusions	48
6	TRANSPARENT OBJECT EXTRACTION FROM REGULAR TEXTURED BACKGROUNDS BY USING BINARY PARALLEL OPERATIONS	49
	6.1 Introduction	49
	6.2 Lighting for Transparent Objects	49
	6.3 Textured Background for a Transparent Object	52
	6.4 Dilatation-Erosion Operations on Binary Images	54
	6.5 Experiments	55
	6.6 Discussion	60

TABLE OF CONTENTS (Continued)

	Page
6.7 Conclusions	62
7 CONCLUSIONS	64
ACKNOWLEDGEMENTS	66
REFERENCES	67
APPENDIX A1. GMAT - INDUCTIVE DYNAMIC PROGRAMMING MATCHING PROGRAM	72

LIST OF FIGURES

1	Correspondence between model and input in dynamic programming process	6
2	Symmetric and nonsymmetric recurrent calculation in the DP process	9
3	Two-dimensional inductive type DP process	12
4	Precedable segments in 2D locally ordered type DP	12
5	Dynamic programming calculation with directional information	13
6	The cyclic DP process	15
7	Allowable changes of the reference model	20
8	Sampled image and extracted rough shape	25
9	Sampled image and the refined result	25
10	Original kidney image under high microscopic magnification, and the superimposed final result corresponding to the highest (first) maximum in f(M, i, j)	26
11	The first four maxima, using gray level	26
12	First four maxima using the directional information	27
13	Sampled image with pitch $\alpha = 8$ and the first five maxima using gray level, $\lambda = 2$, $\rho = 32$, $\beta = 4$, array = 80 × 60	28
14	Original image with low microscopic magnification and first 10 maxima using direction	28
15	Variations in 2D domain caused by 3D tilt	31
16	Block diagram of the process	32
17	SR-0	34
18	SR-1	34
19	Edge amplitude, $\alpha_1 = 1$	35
20	Rings on 80 × 60 image	35
21	Scissors on SR-1	36
22	SR-2	36
23	Edge amplitude	37
24	Results of SR-2	37
25	AT-0	38
26	AT-1	38

LIST OF FIGURES (Continued)

		Page
27	Edge amplitude, $\alpha_1 = 2$	39
28	Histogram equalized AT-1	39
29	Results of AT-1	40
30	Results of AT-2	40
31	Results of AT-3	40
32	Travelling in input segment tree	44
33	The rotational independent DP process	46
34	Image of the ashtray with lighting from top and black background	50
35	Gray background	50
36	White background	51
37	Polarized lighting	51
38	Image of the ashtray with diffused lighting from top	51
39	Diffused lighting from bottom	51
40	Variation of diffused light from bottom, (a) for a convex lens and (b) for a concave lens	52
41	Shadow on the background (graph below) and the observed intensity change (graph above) by (a) convex lens and (b) concave lens	53
42	Schematic layout of the experiment with an ashtray on textured background	56
43	Gray level image a	56
44	b = BIN[85](a)	57
45	c = SMLG[4, 1]	57
46	d = LGSM[4, 5]SMLG[4, 1](b)	58
47	f = SMLG[4, 5]LGSM[4, 1](b)	58
48	g = OR(NOT(d), f)	59
49	h = SMLG[8, 10]LGSM[8, 10](g)	59
50	The ashtray and test sheet with continuous gray levels on square black dot background	60
51	Result for square dot background	60

LIST OF TABLES

1	Dynamic programming in image processing and pattern recognition	8
2	Center of gravity of extracted metal rings (SR-1)	35
3	Rotational center of scissors (SR-1)	36
4	Center of gravity of extracted metal rings (SR-2)	37
5	Rotational center of scissors (SR-2)	37
6	Results for the black dot removal	61
7	Two kinds of "shape from texture"	62

ABSTRACT

Existing applications of Dynamic Programming (DP) to image processing and pattern recognition are reviewed and classified into one-dimensional (1D) DP and two-dimensional (2D) DP. Two-dimensional DP is further classified into 2D neighbor type, 2D inductive type, and 2D locally ordered type.

The 2D inductive type DP is augmented so as to be multi-resolutional, cyclical, and directional. The optimally fitting shape in an input image is extracted, while allowing for a certain amount of length change of each side of a model polygon. The method is applied to extract kidney glomeruli in microscopic images and to recognize relatively flat objects; one group has reflective metal scissors and rings and the other has transparent glass asktrays, both on a relatively unconstrained background and in standard illumination.

A rotation-free, cyclic DP matching method is proposed to recognize 3D objects in range data. The reference models consist of a series of segments where each model constitutes a closed loop. Changes of the length and the relative angle with respect to the preceding segment are evaluated during the DP process. In the last chapter, a regularly textured background is proposed to utilize both the reflection and refraction effects of transparent objects. The observed image is processed by conventional binary local parallel operations, i.e., dilatation (expansion) and erosion (contraction). Both the dilatation-erosion and erosion-dilatation operations are used symmetrically and are unified. The method is threshold insensitive.

RÉSUMÉ

Les auteurs passent en revue les applications actuelles de la programmation dynamique (DP) dans les domaines du traitement des images et de la reconnaissances des formes et structures; ces applications sont réparties en DP unidimensionnelle (1D) et en DP bidimensionnelle (2D). Il existe trois types de programmation 2D: la 2D par proximité, la 2D, inductive et la 2D par classement local.

La programmation dynamique 2D inductive a été enrichie, de manière à produire une méthode multirésolution, cyclique et directionnelle. La forme la mieux adaptée est extraite de l'image d'entrée; il est possible de faire varier dans une certaine mesure la longuur de chaque côté d'un polygone modèle. Cette méthode est appliquée à l'extraction de glomérules rénaux enregistrés sur des images microscopiques et à l'extraction de glomérules rénaux enregistrés sur des images microscopiques et à la reconnaissance d'objets relativement plats; un groupe d'objets contient des bagues et des ciseaux métalliques brillants réfléchissants et l'autre des cendriers en verre transparent. Dans les deux cas, l'arrière-plan est relativement simple et le niveau d'éclairage est normal.

Les auteurs proposent le recours à une méthode de concordance en DP cyclique sans rotation pour la reconnaissance des objets tridimensionnels grâce aux données de distance. Les modèles de référence sont formés d'une série de segments dans lesquels chaque objet constitue une boucle fermée. Les variations de la longueur et de l'angle relatif entre deux segments consécutifs sont évalués pendant l'opération de programmation dynamique. Dans le dernier chapitre, un arrière-plan à texture régulière est proposé pour exploiter les effets de réflexion et de réfraction des objets transparents. L'image observée est traitée par des opérations locales binaires parallèles ordinaires, c'est-à-dire la dilatation (expansion) et l'érosion (contraction). Les opérations de dilatation-érosion et d'érosion-dilatation sont utilisées symétriquement et ont été uniformisées. La méthode est indépendante de tout seuil.

DYNAMIC PROGRAMMING IN PATTERN RECOGNITION

H. Yamada and T. Kasvand

CHAPTER 1. OVERVIEW

This report consists of five relatively independent and self-consistent chapters, and all but the last chapter deal with applications of dynamic programming techniques to image processing and pattern recognition. The Dynamic Programming (DP) method has been used mainly in speech recognition, because it is easy to apply DP to one-dimensional (time-serial) features. There are some difficulties in applying DP to two-dimensional (2D) images.

In Chapter 2, the difficulties of applying DP in 2D image processing and pattern recognition are analyzed, existing methods are reviewed by classifying them into several types, and several aspects of DP methods are expanded.

In Chapter 3, the optimally fitting shape in an input image is extracted by using a polygonal model with variable sides. The method uses multi-resolution to reduce the calculation time, and the information on directionality of contours is introduced for higher performance. The cyclical property of a contour also is taken into account. The method is applied successfully to extract kidney glomeruli in microscopic images.

In Chapter 4, a DP matching method is applied to object recognition. The top views of relatively flat objects are used. The optimally fitting shape is extracted from an input image, while allowing for a certain amount of change of length of each side of a model polygon. Information on directionality of contours is used for high performance and the cyclical property of a contour is also taken into account. The method is applied to two groups of objects: one group contains reflective metal scissors and rings and the other has transparent glass ashtrays. Both are on a relatively unconstrained background and in standard illumination.

In Chapter 5, a model-based 3D object recognition method by using DP is proposed. The input information consists of preprocessed range images where the input edges are approximated by linear segments. The reference models consist of a series of segments where each model constitutes a closed loop and each segment in the model is longer than the corresponding input segment. The optimal correspondence is extracted by using DP, allowing for a certain amount of change of the length and the relative angle with respect to the preceding segment. In the process, one model segment

is allowed to correspond to multiple input segments, but the inverse is not allowed. The cyclical property of model segments is taken into account.

In Chapter 6, a transparent object is investigated from the viewpoint of automatic image analysis and recognition. A regularly textured background is proposed to utilize both the reflection and refraction effects of the object. A transparent ashtray is put onto a dotted background and observed from the top by a TV camera. The dot size of the background is known in advance. The observed image is thresholded and processed by conventional binary local parallel operations, i.e., dilatation (expansion) and erosion (contraction). Both the dilatation-erosion and erosion-dilatation operations detect regions larger than the original dot. These regions are either reflecting areas or areas created by deformed or enlarged dots. These two results, which have a compensatory property, are unified. This method is threshold insensitive and is applicable also to opaque objects which have continuously changing gray levels. The results can be used to improve the performance of DP methods.

CHAPTER 2. DYNAMIC PROGRAMMING IN IMAGE PROCESSING AND PATTERN RECOGNITION

2.1 Formulation of a Matching Problem

The process for computer vision, image analysis, or pattern recognition can be divided into two subprocesses: one is bottom-up feature extraction and the other is matching between input features and reference models (knowledge of the shape). Feature extraction may include representation of the features, and the matching may include learning and retrieval of the knowledge. Feature extraction, representation, and matching are the key problems. As a method for the matching, dynamic programming has been used. In this chapter, how the DP has been applied is reviewed.

Dynamic programming is a mathematical method of solving at high speed the problem of a multi-stage optimal decision process by computer. There, "the principle of optimality" (1) states that "an optimal policy has the property that whatever the initial state and initial decision are, the remaining decision must constitute an optimal policy with regard to the state resulting from the first decision" has an important role. By using this principle, the multi-stage process is divided into a sequence of "recurrence relations" of one-stage optimal decision processes. The main advantages of the resulting DP algorithms are (2):

1. The problem can be solved without knowing in detail the statistical properties of the measured signal.

2. Due to the formulation in terms of optimization, immunity against local disturbances is achieved.

3. The optimal solution is reached after a finite number of computation steps; there are no convergence problems.

4. The algorithm allows a straightforward machine implementation.

In order to illustrate some properties of DP, the formula used in the DP process is described. At first, an unknown input pattern is expressed by

$$[1] \qquad U = \{u[i]\} = \{u_i\}, \quad i = 1, 2, \ldots, I$$

where u[i] or u_i is a feature element, a line segment for example, which is to be extracted from the input image. In the case that the picture element (pixel) itself is used as an input representation (pixel-wise representation), we can rewrite i by (i, j) as,

[2] $$U = \{ u(i, j) \}, \quad i = 1 \sim I, \quad j = 1 \sim J$$

A reference model R which we want to extract from or match with the input is defined by

[3] $$R = \{r[m]\} = \{r_m\}, \quad m = 1, 2, \ldots, M$$

When the model element m corresponds to a set of "ordered" input elements, the set is written as

[4] $$C[m] = \{i_b[m], i_b[m] + 1, \ldots, i_e[m] - 1, i_e[m] = i[m]\}$$

where i_b is the beginning element and i_e is the end element. Then, an evaluation value E, when r[1], r[2], ..., r[M] correspond to input elements C[1], C[2], ..., C[M], respectively, can be written as

[5] $$E(R, C[1], C[2], \ldots, C[M]) = \sum_{m=1\sim M} \{d(m, C[m]) + \sum_{m'<m} e(m', m, C[m'], C[m])\}$$

This equation shows that E is decided by the sum of the following two measures:

1. Local distance d is defined between model element m and corresponding set of input elements C[m].

2. Relative evaluation e is defined between the relative information in the model and the relative information in the input, for example, the difference between a configuration of model elements and those of input elements for measuring the warping due to displacement.

In the case that the relative distance can be measured "only between the preceding element," we can rewrite it by

[6] $$E(R, C[1], C[2], \ldots, C[M]) = \sum_{m=1\sim M} \{d(m, C[m]) + e(i_e[m - 1], i_b[m])\}$$

The matching between the model and the input can be written as the problem to find the optimum value of E, i.e.,

[7] $$D(R, U) = \min_{\{C[1],C[2],\ldots,C[M]\}} E(R, C[1], C[2], \ldots, C[M])$$

If we solve this equation by exhaustive search, then the number of combinations easily becomes nonpermissible (impossibly large) with the increase of input and model elements.

2.2 Introduction of Dynamic Programming

To decrease the calculation, we introduce the DP technique. In the DP multistage decision process, the variable C[m] is thought of as the policy decided in the mth stage, $\{C[m] \mid m = 1, 2, \ldots, M\}$ as all combinations which we can choose as policies at the 1st, 2nd, ..., mth, ..., and Mth stages, and D as the whole evaluation value (criterion function) when the optimal sequence of policies is taken.

In DP calculation, we introduce the intermediate evaluation function f(m, i). This function has the property that "under the assumption that m corresponds to C where the end element of C is $i(=i_e)$, f(m, i) is the optimum value within all possible sequences of $\{C[1], C[2], \ldots, C[m-1], C[m]\}$." That is,

$$[8] \qquad f(m, i) = \min_{\{C\}} \sum_{m'=1\sim m} \{d(m', C[m']) + \sum_{m''<m'} e\{m'', m', C[m''], C[m']\}\}$$

When the evaluation E can be written as Eq. [6], Eq. [8] is expressed by a recurrent form such as

$$[9] \qquad f(m, i) = \min_{\{C\}} \sum_{m'=1\sim m} \min_{i'\in P(i_b)} \{f(m', i') + e(n', m, i', C) + d(m, C)\}$$

This recurrence equation determines the policy and the evaluation value at the mth stage under the condition that the policy and the evaluation values have been already decided with respect to the 1st, 2nd, ..., (m - 1)th stages. The equation comes from "the optimal principle" of "the optimal policy at mth stage does not depend on the former decision process (before m'), but is decided only by the state (evaluation value) at m'." This decomposition to the recurrence relation is the key to dynamic programming (see Fig. 1).

Here, the set P specifies input elements which can correspond to model element m' under the condition that m corresponds to C. Note here that the introduction of the concept of this "precedable" set P will play an important role in the following classification of existing DP techniques and expansions to more "two-dimensional" DP. This concept comes from ref. 3.

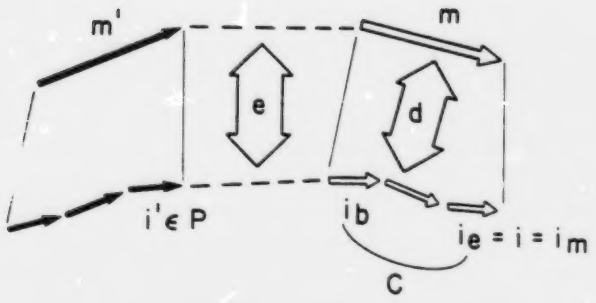


Figure 1. Correspondence between model and input in dynamic programming process. Suppose m corresponds to i_b to i_e. Then, the evaluation at (m, i_e) is measured by the summation of: (1) local distance d between m and C, (2) transition difference e between (from m' to m) and (from i' to i_b), (3) the preceding evaluation value up to (m', i'), i.e., f(m', i').

The DP calculation for solving Eq. [9] is done as follows: At the first stage of m = 0, we set the initial value such that

$$[10] \qquad f(0, i) = 0$$

Then, the calculation is done recurrently at each stage of m = 1, 2, ..., M. At stage m, the recurrence calculation is

$$[11] \qquad f(m, i) = \min_{i_b} \min_{i' \in P(i_b)} \{f(m - 1, i') + e(i', i_b) + d(m, C)\}$$

where m' is restricted to m - 1, and the displacement from the end point of m' to the beginning point of m always equals zero. In this case, P is defined with respect to each i_b and e is measured as the gap from i' to i_b. The $e(i', i_b)$ term is a measure of the beginning i_b of section m and the prior optimal end point i' of section m - 1. In the special case that the displacement e need not be evaluated, we can write the recurrence relation as

$$[12] \qquad f(m, i) = \min_{i' \in P} \{f(m - 1, i') + d(m, i', i)\}$$

In the pixel type of DP calculation, C can be sometimes be rewritten in terms of the points on the line from (i', j') to (i, j) as

$$[13] \qquad f(m, i, j) = \min_{(i', j') \in P} \{f(m - 1, i', j') + d(m, i', j', i, j)\}$$

In this case, the factor e is omitted, and i' and i_b in Eqs. [11] or [12] is identified.

After the calculation of the recurrence relation for m = 1, 2, ..., M, the global evaluation value is decided as

[14] $$D(R, U) = \min_{\{i\}} f(M, i)$$

Here, this solution is optimal. That is, the value D(R, U) is identical to the D of Eq. [7].

After these preliminaries, let us classify the DP used in the pattern recognition area as in Table 1. Note that M or I in the column of the number of calculation is dependent on the representation of the element, that is, when the representation is pixel-wise then the number is much larger than the case of feature-wise representation.

2.3 Monotonic Continuous Dynamic Programming in Image Processing and Pattern Recognition

In the area of pattern recognition, DP was at first used to recognize speech (19, 20, 27). There, DP is used to get the optimal correspondence and evaluation after the nonlinear warping of the time scale. Now, it can be said that DP is the standard technique for matching for speech recognition.

In the case of a two-dimensional image, if the feature is time-serial like on-line character recognition or a dynamic scene, we can apply DP in the same way as in speech along the time sequence (4-6, 28-31). Even in off-line character recognition, one-dimensional DP can be applied to the partial sequence of features within which an ordering can be assumed (7, 8), to the projected one-dimensional features (9, 10), or to some restricted features (32). In the case of stereo matching, if the scan lines of the camera are made coincident with stereo disparity, the problem can be separated on each scan line, so one-dimensional DP can be applied effectively (23-26).

In all these cases, the problem is solved under the condition that both input and model features can be ordered and there is no disorder in either of them. That is the monotonic continuous condition. In the recurrence relation, m' is only m - 1, and the range of i' is restricted to some continuous predecessors of i. For example, the region of i' in Eq. [12] is a set like

[15] $$P(i) = \{i'\} = \{i - 1, i - 2, i - 3, i - 4\}$$

Table 1. Dynamic programming in image processing and pattern recognition

DP type[1]	Input element	Shape model	Hierarchy	Calculation[2]	Two-dimensionality[3]	Application (References)
1DM	Feature	Feature	No	MI =4	M	Online character (5, 8)
1DM	Feature	Feature	Model	CMI=5	M,G	Online character (6)
1DM	Feature	Part	No	-	M	(7, 8)
1DM	Project	1D	No	CMI=5		(9, 10)
1DC	Point	Point	No	MI =4		(11)
2DI	Line	Pixel	No	CMI=7	L,M	(12, 13)
2DI	Line	Edge	Input	CMI=7	L,M	(14)
2DD	Line	Line	No	M´I=22	M	(15)
2DL	Line	Line	Model	CMI=5	M,G	(3)
2DG	Feature	Yes	No	CMI=7	G	(16)
2DF	Pixel	No	No	CI	L	Feature extraction (17)
2DH	Pixel	No	No	CI	L	Smoothing (18)
1DM	Feature	Yes	No			Speech (19, 20)
1DM	Feature	Yes	Model			Speech (21, 22)
1DM	Letters	No	No			Text segmentation (12)
1DM	Pixel	Another	No			Stereo (23, 24)
1DM	on a	row	Line			Stereo (25)
1DM	row		Input			Stereo (26)

[1] Type of DP: 1DM = Monotonic Continuous, 1DC = Continuous Cyclic, 2DN = 2D Neighbor, 2DF = 2D Full Neighbors, 2DH = 2D Half neighbors, 2DI = 2D Inductive, 2DL = 2D Locally Ordered, 2DD = 2D Disordered, 2DG = 2D Graphical.

[2] Number of calculations: M = # model elements, I = # input elements, C = constant (= 10). Note M or I is dependent on the representation of the element. Number in the table indicates the power of 10, when pixels = 100, features = 10000.

[3] Two-dimensionality: L = local, M = middle, G = global.

Sometimes the following equation is used instead of Eq. [15].

$$[16] \qquad f(m, i) = \min \begin{cases} f(m-1, i-1) + d(m, i) \\ f(m-1, i-2) + d(m, i-1, i) \\ f(m-2, i-1) + d(m-1, m, i) \end{cases}$$

This equation is for symmetrical calculation between m and i. However, if input features are forced to be cut in two, the equation can be converted to Eqs. [12] and [15]. So, almost all of the arguments are done by using the unsymmetrical form of the recurrence relation (see Fig. 2).

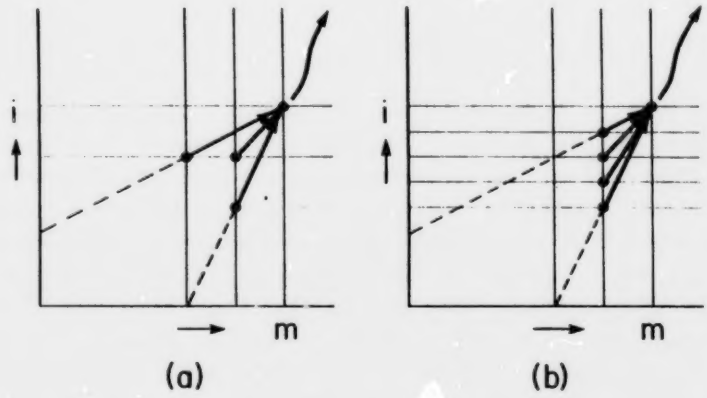


Figure 2. Symmetric and nonsymmetric recurrent calculation in the DP process. The nonsymmetric DP makes the formulation simple, and the symmetric scheme (a) can be transformed into the nonsymmetric scheme (b) without loss of generality by increasing input sampling pitch.

Let us call this the "monotonic continuous" type of DP. In applications of DP to pattern recognition, this type has been used in an overwhelming majority of cases. The monotonic continuous DP has fewer calculations and, as stated, it can be applied to the special problems even in the case of two-dimensional images. However, generally speaking, it has a drawback for two-dimensional objects in that it cannot fully evaluate the two-dimensional information. In the true application of DP to a two-dimensional image, the key problem is how to get rid of this one-dimensionality of the DP.

2.4 Two-dimensional Neighbor Dynamic Programming Without Model

If we can say that the DP for one-dimensional data is of the monotonic continuous type, then it follows naturally that we should

take two-dimensional neighbor of i as the range of i', for two-dimensional data. Now, let us call this the "two-dimensional neighbor" type.

Montanari (33) described the equivalence between the distance transformation and the decision of the minimum path by DP. Dynamic programming was also applied to the problem of the extraction of a smooth curve in a noisy picture (2, 17, 34) and for smoothing of the features in an image (18). In all these cases, DP was formalized as a 2D neighbor type operation. Equation [13] is rewritten as

$$[17] \qquad f(i, j) = \min_{(i',j')\in P} \{f(i', j') + d(i', j', i, j)\}$$

where (i', j') satisfies

$$[18] \qquad \max \{|i' - i|, |j' - j|\} = 1$$

We can say that this is "the most" two-dimensional in the sense that the 2D neighbors are used in the method. However, note here that there is no m in the equation. That means, in all cases, that there was no "model" of the shape of the object. There is only the appearance of global evaluation; for example, in the case of ref. 17, only the minimum distances between two points are evaluated. In that sense, it can be said that the DP is not being applied to the matching problem but for feature extraction. In Table 1, 2D neighbor type DP is divided further into two types: 2D full neighbors and 2D half neighbors. The full neighbors mean all surrounding neighboring pixels, and the half neghbors mean left and upper part only.

The difficulty of introducing the shape of the model in the 2D neighbor type is caused by the following:

1. When we permit the use of 2D neighbors, the range of variation becomes too large and the result becomes too loose.

2. It becomes difficult to decompose the evaluation process into a series (recurrence relationship).

However, in the case that there is a strong limitation on each corresponding point, we can apply this scheme to two-dimensional images (34, 35).

2.5 Two-dimensional Inductive Dynamic Programming on Input Pixel

As the other way to solve the problem stated, Kovalevsky (12) and Sakoe (13) gave a sequence to a reference model, and also the range of input neighbor was controlled by the induction of the model. That is, in

the recurrence relation of Eq. [12], the range of (i', j') is changed to meet a demand on (m, i, j) (see Fig. 3). If

$$|\cos t_m| > |\sin t_m|$$

then

$$[19] \qquad P = \{(i', j') \mid i' \in [i - s_m/\lambda \cdot \cos t_m,\ i - \lambda \cdot s_m \cdot \cos t_m],\ j' = (i' - i) \tan t_m - j\}$$

If

$$|\cos t_m| < |\sin t_m|$$

then

$$[20] \qquad P = \{(i', j') \mid j' \in [j - s_m/\lambda \cdot \sin t_m,\ j - \lambda \cdot s_m \cdot \sin t\ ],\ i' = (j' - j)\cot t_m - i\ \}$$

The set P is defined (induced) by (i, j) and s_m at the calculation of the recurrence function, so we call this the 2D inductive type DP.

The frame of this type was so simple that it should have a wide range of applications. However, it has not been applied afterwards because the number of calculation becomes large in this scheme. In Chapters 3 and 4 a multi-resolutional scheme is used to avoid the increase of the calculations. There, directional information is introduced to improve the performance, and the cyclic property of the model (contour) is taken into the evaluation function.

2.6 Locally Ordered Dynamic Programming on Feature Element

To decrease the number of calculation and to increase the performance, Yamada (3) proposed feature extraction before DP processing. The extracted feature components were used as the input elements of the DP calculation, and the recurrence relation is

$$[21] \qquad f(m, i) = \min_{i'} \min_{i'' \in P(i')} \{f(m - 1, i'') + e(i'', i') + d(m, i', i)\}$$

There, by introducing the local interdependency (local ordering) in input features as the "precedable segments," high speed of calculation is achieved while retaining the performance (see Fig. 4). Actually, the formula used there was

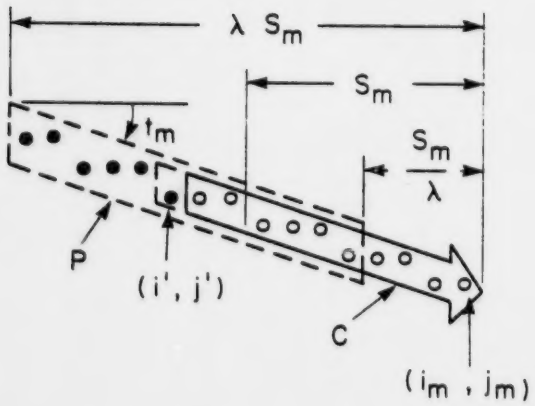


Figure 3. Two-dimensional inductive type DP process. In order to calculate the DP step m, at first an arbitrary point (i_m, j_m) is fixed as an end point of segment m, then, the region P of the beginning points is calculated from the angle t_m of the segment and the extent of variation λ. Second, an arbitrary point (i', j') on P is fixed as the beginning point, which fixes the region C along the line from (i', j') to (i_m, j_m). Then we can calculate the local distance $d(i', j', i_m, j_m)$ as the averaged gray levels in C. At this time, the evaluation value up to m under the condition that the end point is fixed to (i_m, j_m) and the beginning point is fixed to (i', j') is $f(m - 1, i', j') + d(i', j', i_m, j_m)$. The maximum value of this f + d with respect to (i', j') in P is the decision with respect to (i_m, j_m), that is, $f(m, i_m, j_m)$.

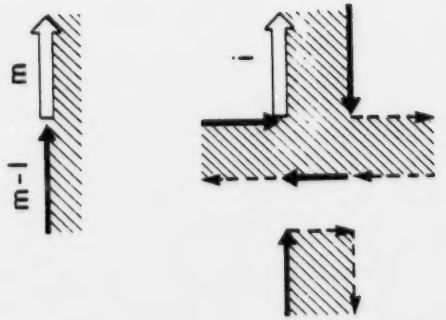


Figure 4. Precedable segments in 2D locally ordered type DP. When m corresponds to i, m - 1 can correspond to five input segments shown by the heavy solid arrows. These can be registered in advance with respect to each i as those have a local minimum distance to i.

$$[22]\quad f(m, i) = \min \begin{cases} \min\limits_{i'' \in P(i-1)} \{f(m-1, i'') + e(i'', i) + d(m, i)\} \\ \min\limits_{i'' \in P(i-2)} \{f(m-1, i'') + e(i'', i-2) + d(m, i-1, i)\} \\ \min\limits_{i'' \in P(i-1)} \{f(m-2, i'') + e(i'', i-1) + d(m-1, m, i)\} \end{cases}$$

However, as stated, this is equivalent to the former equation, if input segments are divided into two pieces.

2.7 Further Expansions

We have reviewed the applications of DP to pattern recognition and classified them into several categories. Now, let us look at other important aspects when DP is applied to a pattern recognition problem, especially to a matching problem.

2.7.1 Directional Evaluation in Dynamic Programming

In ref. 3, the directional information of each contour segment has also been used in the locally ordered type DP. It has an important role in maintaining high performance. Yamada et al. (36), also applied 2D inductive type DP to ultrasonic kidney image and had a good result by using the directional information. According to these, the information of direction is an important property for image processing. A similar scheme is used in Chapters 3 and 4 (see Fig. 5). Evidence that directional information is

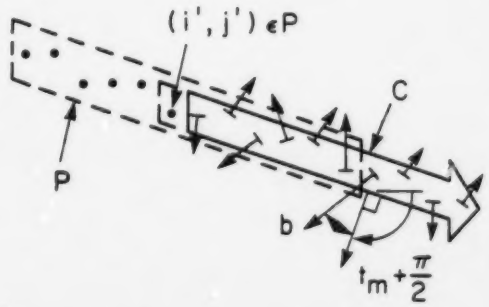


Figure 5. Dynamic programming calculation with directional information. Local distance in the case that the beginning and the end points are fixed to (i', j') and (i_m, j_m), respectively, is measured as the difference between the angle of the edge operator and the right angle to the model segment. Besides, the gray level itself can also be considered in the case where the difference angle is large.

effective for vision processing may be found from physiological studies of biological vision systems (37).

2.7.2 *Cyclic Dynamic Programming*

When one-dimensional DP is applied to a cyclic model, it is difficult to evaluate the distance between the beginning point (x_0, y_0) and the end point (x_M, y_M), i.e., is it short or not. This is a typical problem illustrating the inability of using two-dimensional information in the DP evaluation process. To improve the equation, Yamada et al. (36) introduced two aspects. One is the evaluation of the distance between the beginning point (x_0, y_0) and the end point (x_M, y_M), after the M stage recurrence process. That is, instead of Eq. [14], let

$$[23] \qquad D(R, U) = \min_{(i,j)} [f(M, i, j) - \gamma \cdot \{|B_x(i, j) - i| + |B_y(i, j) - j|\}]$$

where $(B_x(i, j), B_y(i, j))$ is the beginning point of r_1 when we fixed the end point of r_M as (i, j). B_x and B_y can be obtained by the trace-back of the former $M \times I \times J$ matrix. Here γ (=1) is a constant for weighting the gap between the beginning point of the beginning segment and the end point of the end segment.

As the second improvement for the coincidence, calculations for more than one cycle are introduced (see Fig. 6). The initial value of f for m = M' (=[M/2]) is set to zero, i.e.,

$$[24] \qquad f(M', i, j) = 0$$

The recurrent calculation is then done with respect to m = M' + 1, M' + 2, ..., M - 1, M, 0, 1, 2,..., M. The evaluation value at m = M at the first time (from M' + 1 to M) is retained as

$$[25] \qquad f_0(i, j) = f(M, i, j)$$

At the final stage, the global evaluation value is obtained from

$$[26] \qquad F(M, i, j) = \min_{(i,j)} [f(M, i, j) - f_0(B_x(i, j), B_y(i, j)) - \gamma \cdot \{|B_x(i, j) - i| + |B_y(i, j) - j|\}]$$

These two schemes are applicable to both one- and two-dimensional-type DP.

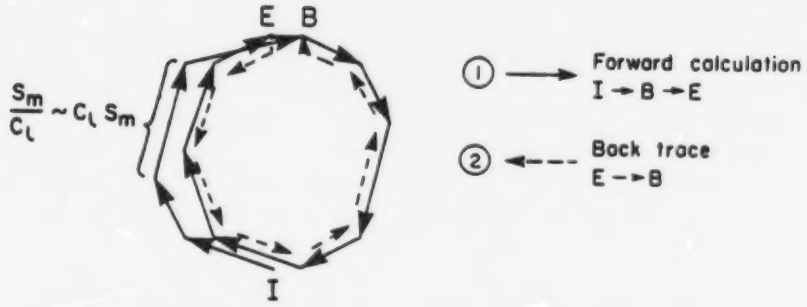


Figure 6. The cyclic DP process. (1) Forward process begins at I. Note that here I is not fixed at one certain point. It is assumed to be at every place in the image. (2) At B after half a cycle of forward calculation, results are saved temporarily. (3) At E after one and a half cycles, the forward calculation ends, and the matrix made by the forward calculation is traced back toward B. (4) Total evaluation value is measured as the averaged value along the sides from B to E, minus the difference in distance between B and E. This evaluation is done with respect to all the points assumed to be E.

2.7.3 Hierarchical Dynamic Programming - Hierarchy in Model Structure

Generally speaking, the global evaluation function in a two-dimensional problem relates to all the features in a two-dimensional space and each feature depends upon the other. If we evaluate this as it is, the computation becomes an exhaustive combination of relations, such as in Eq. [5]. When we apply DP to this problem, we must cut off the branches of the search which are not so effective for the global evaluation (16). However, if we cut off the branches too much and make the problem one-dimensional, then 2D global evaluation becomes difficult.

One of the effective ways to keep the balance between local evaluation and global evaluation is a hierarchization of DP. In speech recognition, the various levels of the problem, such as phonetic level, word level, and sentence level, exist together at one time, and they cannot be divided so easily. To solve this situation, DP was applied hierarchically to each level, and optimal resolution for global evaluation was obtained (21, 22).

In the case of a 2D image, a hierarchical scheme has also been used in the monotonic continuous case, as in on-line character recognition, to absorb the variations in the sequence of writing and in the number of touches (strokes) (6), and in stereo-matching when using the right-angle information to the scan line (across scan lines) in the matching (25).

In ref. 3, the hierarchy was used in the 2D locally ordered type DP matching. At first, the low level evaluation is calculated with respect to each cycle of each contour in a model. The evaluation value at the end of the contour l (l = 1, 2, ..., L) is expressed by $f_\lambda(m_\lambda, i)$, where f_λ is the recurrent function for contour l and m_λ is the last model segment in l. The relation between contours are linked by using the minimum spannin tree, where the node of the graph is the end point of each contour and the length of the edge is defined by the Euclidean distance between two end points. The high level DP calculation begins from deeper nodes. That is, suppose the high level recurrence function at the contour l is F(l, i), and l_1, l_2, ..., l_k, ..., are contours deeper by one than l, then the recurrence calculation is done by

$$[27] \qquad F(l, i) = f_\lambda(m_\lambda, i) + \sum_{\{l_k\}} \min_{i'} \{F(l_k, i') + e(l_k, l; i', i)\}$$

where e is the difference of the relative distances, i.e., the difference of (1) distance between end points of two model contours and (2) distance between i and i'. By calculating this sequentially from deeper contours, the evaluation value for the whole tree is given by

$$[28] \qquad D(R, U) = \min_i F(l_{root}, i)$$

where l_{root} is the number of root contours. Consequently, an input image can be evaluated two dimensionally.

In ref. 16, DP was applied to decide the optimal matching of sub-features of human face, like eyes, a nose, a mouth, etc., by using the result of template matching of sub-features at all positions. This corresponds to the higher hierarchy in ref. 3. However, the intrinsic part of this method is that the pixel-wise representation is used for both input and model. It would be better to classify this approach into another category, for example, 2D graphic type DP.

2.7.4 Multi-resolutional Dynamic Programming - Hierarchy on Input Resolution

The another type of hierarchization is for input resolution. That is, hierarchy by multi-resolution. This is used for reducing computation in stereo matching (26). In ref. 36, at first, the sampling at pitch α from U of I × J matrix to U' of K × L (K = I/α, L = J/α) is done as,

$$[29] \qquad u'(k, l) = \sum_{(l-1)\alpha<j<l\alpha} \sum_{(k-1)\alpha<i<k\alpha} u(i, j)/a^2$$

The DP calculations are done as before except that the input image is u'(k, l), a matrix of size K × L.

After processing on the K × L plane, the correspondence points $\{(x'_m, y'_m)\}$ are put back into the corresponding positions $\{(x_m, y_m)\}$ on the I × J plane, that is,

$$x_m = \alpha \cdot x' - (\alpha - 1)/2$$
$$y_m = \alpha \cdot y' - (\alpha - 1)/2 \qquad [30]$$

After this, the same DP calculation is repeated with respect to the plane U(I × J), but under a constraint for each end point. In this case, the calculations are reduced due to the constraints on the positions. In total, the calculations for the first stage decrease to $1/\alpha^3$, and for the second stage decrease to $(2\varepsilon)^2/(I \times J)$, where ε is the allowance for the position. In the present case, it is used to reduce the input resolution. In ref. 39, it is reported that the reduction of resolution before feature extraction makes the directional information clearer.

In both cases of hierarchical DP and multi-resolutional DP, we can say that the hierarchy rearranged the recognition process where previously the problems of different levels existed simultaniously and made the total efficiency high. By arranging the problem as multi-level, and by interchanging the information between different levels, we can get a well-balanced global evaluation.

CHAPTER 3. RECOGNITION OF KIDNEY GLOMERULUS BY INDUCTIVE DYNAMIC PROGRAMMING MATCHING

3.1 Introduction

This chapter concerns the extraction of shapes in microscopic images of kidney tissue sections. The objects to be recognized (glomeruli) each consists of a black membrane forming a more or less circular structure. Within the object there is a black mass occupying most of the area. Surrounding the mass and next to the outer membrane is a white ring which is fairly smooth but varying in thickness. The mass itself is of varying density and is separated by white channels connecting the white ring. An example is shown in Figs. 10 and 14. The ultimate objective is to measure the area of each glomerulus and the area enclosed by the white ring. It is expected that the size distribution of the glomeruli and the ratio of the black mass to the total area are affected by the treatment of kidney tissue with certain drugs.

In the area of biomedical image analysis, such as the present one, the observed image is noisy, since we cannot create an "ideal" environment for observations. In these conditions, it is difficult to distinguish between the object and the background or to extract the object by using only some bottom-up techniques, such as image enhancement, edge detection, region segmentation. Here, we need some top-down knowledge, i.e., a model of the shape. By using this information, more objects are extracted or recognized. However, there is another difficulty besides the noise in the recognition of these kinds of biomedical objects, i.e., the shape of the object is variable. A strong matching method with input noise rejection and with variation of shape must be developed.

In this chapter, an inductive DP matching method is used. The input representation is a digital image matrix as in refs. 12 and 13. However, to increase the performance, directional information and hierarchy are taken into account, as in ref. 3. The reference model is a polygonal approximation of a contour. The cyclical property of the contour is used here. A similar method for recognizing objects in ultrasonic images was used by Yamada et al. (36). Here the method is applied to microscopic images of kidney tissue sections.

3.2 Inductive Dynamic Programming Matching Method

The gray level input image U is

$$[31] \qquad U = u(i, j), \quad 1 \leq i \leq I, \quad 1 \leq j \leq J, \quad 0 \leq u \leq 1$$

and the reference model, which we use to extract objects from the input image, is

[32] $$R = (r_1, r_2, \ldots, r_m, \ldots, r_M)$$

where R is a sequence of M line segments (vectors), the length of the line segment r_m is $s_m = |r_m|$, and the direction is t_m. Here, we do not assume that the beginning point of r_1 and the end point r_M are in the same position.

We extract the optimal combination of segments (vectors) $\{r_m \mid m = 1 \sim M\}$ and the position (i_M, j_M) of the end segment, which maximize the similarity function S(R, U) when the lengths $|r_m|$ are varied within a certain range (see Fig. 7). This optimal combination can also be expressed as the beginning point (i_0, j_0) of r_1 and the end point (i_m, j_m) of each segment r_m, m = 1, 2, ..., M. As stated, the point (i_0, j_0) and (i_M, j_M) are not necessarily coincident. The optimum is expressed by the formula,

[33] $$S(R, U) = \max_{\{(i_m, j_m), m=0\sim M\}} \sum_{m=1\sim M} d(i_{m-1}, j_{m-1}, i_m, j_m)$$

Here, $d(i_{m-1}, j_{m-1}, i_m, j_m)$ is the local similarity between U and r_m, where the beginning point of r_m (= the end point of r_{m-1}) is (i_{m-1}, j_{m-1}), and the end point of r_m is (i_m, j_m). The local similarity is defined as follows (see Fig. 3, where $i_{m-1} = i'$ and $j_{m-1} = j'$):

[34] $$d(i_{m-1}, j_{m-1}, i_m, j_m) = \sum_{(i'',j'')\in E} u(i'', j'') \Big/ \sum_{(i'',j'')\in E} 1$$

where, if

$$|i_m - i_{m-1}| > |j_m - j_{m-1}|$$

then

[35a] $$E = \{(i'', j'') \mid i'' \in (i_{m-1}, i_m],$$

$$j'' = (i'' - i_m)(j_m - j_{m-1})/(i_m - i_{m-1}) + j_m\}$$

if

$$|i_m - i_{m-1}| < |j_m - j_{m-1}|,$$

then

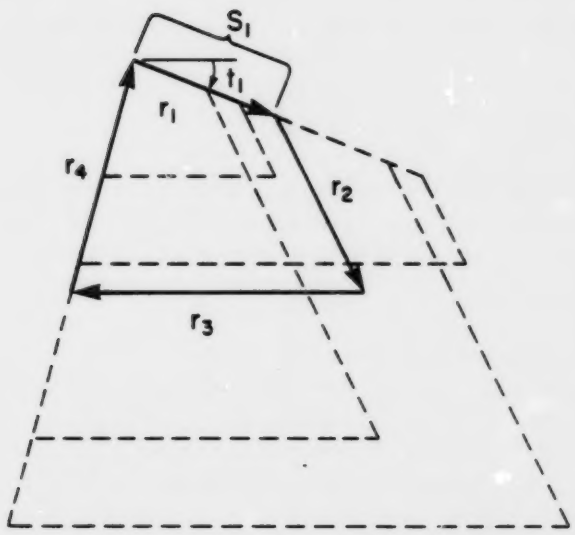


Figure 7. Allowable changes of the reference model. The solid lines show the shape of the standard (original) reference model, a polygon. By changing the length of each edge of the polygon, the resulting shape changes as shown by the dashed lines. The proposed DP matching method finds the optimally fitting object from an input image, both in position and shape, within certain changes of edge lengths. Here the "optimal" means that the averaged gray level along the edges is maximum. In the case of this figure, the extent of the change of any side in the polygon is half to twice, that is, $\lambda = 2$. In the experiments described in this chapter, a regular 16-sided polygon is used as a standard reference model.

[35b] $$E = \{(i'', j'') \mid j'' \in (j_{m-1}, j_m],$$

$$i'' = (j'' - j_m)(i_m - i_{m-1})/(j_m - j_{m-1}) + i_m\}$$

E is the set of points on a line which connects points (i_{m-1}, j_{m-1}) and (i_m, j_m), not including the point (i_{m-1}, j_{m-1}). In the case where the angle to the x-axis is between $-\pi/4$ and $+\pi/4$, then the points are selected with respect to each integer point on the x-axis. If the absolute angle is greater than $\pi/4$, then the points are selected along the y-axis. The value d is the average gray level along the line.

In order to maximize the value S(R, U), we use the DP technique for the M-stage decision processes. As an initial value, we set

[36] $$f(0, i, j) = 0$$

Next, we calculate the recurrence value f(m, i, j) at each input point (i, j) as

$$[37] \qquad f(m, i, j) = \max_{(i',j')\in P} \{f(m - 1, i', j') + d(i', j', i, j)\}$$

where P is an admissible region for the beginning point when the end point is fixed at (i, j). This is defined by Eqs. [19] and [20] (also see Fig. 3). In order to make understanding easier, the operational direction of the function d of Eqs. [33] and [37] was from (i_{m-1}, j_{m-1}) to (i_m, j_m). In actual calculations, it is done from (i_m, j_m) to (i', j'). Consequently, we need not calculate independently d(i', j', i, j) with respect to (i', j'). Hence, f(m, i, j) is the optimal evaluated value with respect to $r_{m'}(m' = 1, 2, \ldots, m)$ where the length of each segment is varied within the range from $1/\lambda$ to λ under the assumption that the end point of r_m is fixed to (i, j).

After calculating the recurrence relation with respect to m = 1, 2, ..., M, the global similarity S can be obtained by,

$$[38] \qquad S(U, R) = \max_{(i,j)} f(M, i, j)$$

Let the (i, j) which maximizes the f be (x_M, y_M), that is,

$$[39] \qquad (x_M, y_M) = \arg \max_{(x,y)} f(M, x, y)$$

The position (x_m, y_m) of the end point of each line segment r_m can be obtained from the back-trace of the I × J × M matrix which contains the selected correspondences within three terms, obtained during the forward recurrence calculations. In the case that each point has a constraint on positions, as stated in Section 2.7.4, the size of the matrix can be reduced to $(2\varepsilon + 1) \times M$, where ε is a constraint constant.

The above mentioned method is the same as the method of Kovalevsky (12) and Sakoe (13) except for the following points:

1. The gray level image is used instead of the binary image.

2. The evaluation is done with respect to the averaged gray level within the corresponding segment. When the sum of the binary pixels was used, the resultant shape tended to become large.

3. Continuous angle for reference model segment is permitted. In the two previous papers (12, 13), only eight directions were allowed; hence, a long line having angle 22.5° could not be extracted.

4. Cyclic DP - Coincidence of beginning point and end point. In order to evaluate the distance between the beginning point (x_0, y_0) and the end point (x_M, y_M), we introduce the Cyclic DP stated in Section 2.5.4.

5. Constraint on shape and position.

6. Multi-resolutional DP.

7. Evaluation of direction (of edge features).

8. Finding of many objects in an image, instead of only the best one.

The last four improvements listed above are explained in the following sections.

3.3 Expansions of the Dynamic Programming

3.3.1 Constraint on Shape and Position

Until now, we have been treating the problem under the assumption that there is no information about the location of each segment. However, if we can get information about where the segments are, at least approximately, then the calculations may be reduced. This case is effective on

1. the problem at time t + 1, under the condition that the result at time t is known

2. the problem of the 3D coordinate z + 1, under the condition that the result of z is known

3. the problem where the approximate shape and position has been given interactively

In these cases, we can constrain each position of s_m by

$$|i - x_m| < \varepsilon, \quad |j - y_m| < \varepsilon \tag{40}$$

and include the knowledge of the shape by:

$$s_m = \left(\left\{(x_m - x_{m-1})^2 + (y_m - y_{m-1})^2\right\}\right)^{1/2} \tag{41}$$

The constraint for the position effectively reduces the calculations, and the knowledge of the shape improves the result.

3.3.2 *Multi-resolutional Dynamic Programming*

If the number of model segments is M, the input picture contains I × J pixels, and the allowance range for segments is Q, then the number of calculations is of the order $I \times J \times Q^M$. By using DP, this is reduced to M × I × J × Q, but this number is still too large. To reduce the calculations, instead of the input gray level itself, the extracted line segment feature was used by ref. 3. By doing this, the number of points (or elements) was reduced to the order of $M \times K^2$, where K is the number of feature elements. Also by using the directional information, the performance was increased. There Yamada (3) also introduced some local dependency among the input segments called "the precedable segments" and succeeded in reducing the number of calculations to the order of M × K.

Here, we also try to reduce the number of points of input, but in this case, we cannot extract an effective line segment at each pixel. Hence, we can only reduce the input points by sampling.

At first, the sampling at pitch α from U of I × J matrix to U' of K × L (K = I/α, L = J/α) is done as

$$[42] \qquad u'(k, l) = \sum_{(l-1)\alpha<j\leq l\alpha} \sum_{(k-1)\alpha<i\leq k\alpha} u(i, j)/\alpha^2$$

The DP calculations are done as before except that the input image is u'(k, l), a matrix of size K × L.

After processing on the K × L plane, the correspondence points $\{(x', y')\}$ are put back into the corresponding positions $\{(x_m, y_m)\}$ on the I × J plane, that is,

$$[43] \qquad \begin{aligned} x_m &= \alpha x'_m - (\alpha - 1)/2 \\ y_m &= \alpha y'_m - (\alpha - 1)/2 \end{aligned}$$

After this, the same DP calculation is repeated with respect to the plane U(I × J), but under a constraint for each end point. In this case, the calculations are reduced due to the constraints on the positions. In total, the calculations for the first stage decrease to $1/\alpha^3$, and for the second stage decrease to $(2\varepsilon)^2/(I \times J)$. In ref. 3, hierarchy was used on the reference model. In the present case, it is used to reduce the input resolution.

3.3.3 Evaluation of Direction

After the 3 × 3 Sobel edge operator, we get the normalized edge strength a(i, j) (≤1) and the edge direction b(i, j). In the case of the sampled plane of size K × L, we use the sampled values of;

$$[44] \qquad \begin{aligned} a'(k, l) &= \max_{(l-1)\alpha<j\leq l\alpha} \max_{(k-1)\alpha<i\leq k\alpha} a(i, j)/\alpha^2 \\ b'(k, l) &= b(i, j) \end{aligned}$$

where

$$[45] \qquad (i, j) = \arg\max_{(i,j)} \max_{(l-1)\alpha<j\leq l\alpha} \max_{(k-1)\alpha<i\leq k\alpha} a(i, j)$$

In the calculation of the recurrence relation, we use the local similarity:

$$[46] \qquad d(k', l', k, l) = \frac{\sum_{(k'',l'')\in E} \max\{\delta u', a'(1 - |b' \sim (t_m + \pi/2)|/2\pi)\}}{\sum_{(k'',l'')\in E} 1}$$

where, u', a', b', represent u'(k", l"), a'(k", l"), and b'(k", l"), respectively (see Fig. 5). The first π/2 comes from the difference between the model direction along the line segment and the input direction given by the edge operator. The angle diference, ~, is calculated with mod 2π, and the results is from -π to π. δ is a weighting factor between gray levels and angles. In the case of the original plane of size I × J, a' and b' are rewritten as a and b, respectively.

3.3.4 Finding Many Objects in an Image

After the forward recurrence calculation in DP, we get the evaluation value with respect to each point under the condition that that point is the end point of the extracting shape. We get the next maximum evaluated point except for the first one by searching for the next maximum point in the evaluation matrix. In that case, we must block the region around the first maximum point by;

$$[47] \qquad f(M, i, j) = \begin{cases} 0, & \text{if } \max\{|i - x_m|, |j - y_m|\} < \beta \\ f(M, i, j), & \text{otherwise} \end{cases}$$

In this way, we get the subsequent maxima and the corresponding shapes.

3.4 Experiments

The method was applied to the extraction of glomerulus structures from microscopic kidney images. Two original images were used. The first image was observed under a resolution of 0.44 μm/pixel (see Fig. 10), the second image at 1.16 μm/pixel (see Fig. 3); both were taken through a 520 nm filter. The size of each is 480 × 640 with 256 gray levels. The gray level was normalized to the range 0 to 1 (see Eq. [31]), thus the actual pixel values are 0, 1/255, 2/255, ..., 1. Here, the multi-resolutional scheme stated in Sections 2.7.4 and 3.3.2 is used (see Figs. 8 to 10).

The reason that the upper-left structure was not extracted in Fig. 11 is that this structure has a long straight side on its right. The method could not cover this amount of variation of the length for one side of the

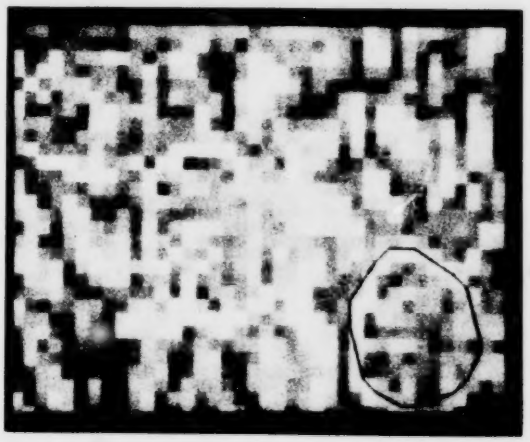


Figure 8. Sampled image and the extracted rough shape. Sampling pitch $\alpha = 16$, array = 40 × 30. The averaged gray level along the sides of the polygon is the maximum in this input image within a change in length of each side from a half to twice ($\lambda = 2$). There is no limitation of position ($\varepsilon = \infty$). The radius of the model $\rho = 80$. CPU time = 27.6 s (VAX11-780).

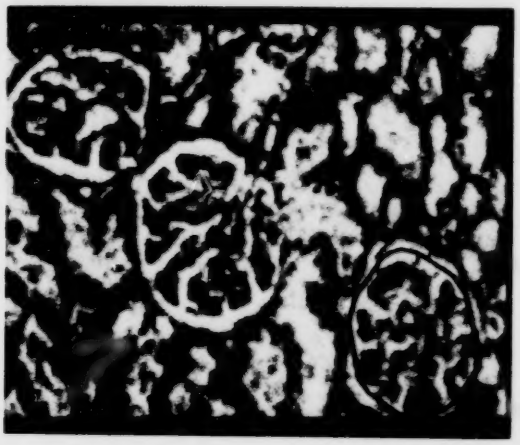


Figure 9. Sampled image and the refined result. Pitch $\alpha = 4$, array = 160 × 120. The position of each vertex in the polygon is constrained within an area no greater than ε (=4) from the result of Fig. 11. In this case, the extent of the variation in the length of a side is limited to $\lambda = 1.2$. CPU time = 14.4 s.

Figure 10. Original kidney image under high microscopic magnification, and the superimposed final result corresponding to the highest (first) maximum in f(M, i, j). The DP calculations are based on gray levels, as Eq. (3), constrained by the results of Fig. 12. Array = 640 × 480, 256 gray levels, $\varepsilon = 4$, $\lambda = 1.2$. CPU time = 183 s.

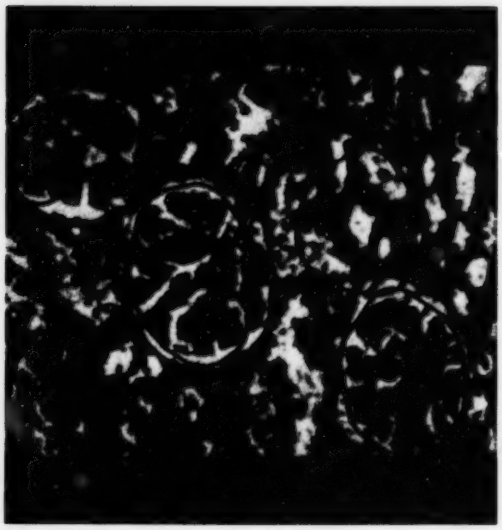


Figure 11. The first four maxima, using gray level. Because the evaluation value has been obtained at each point after the forward DP calculation, several structures in the image can be obtained as the succeeding next maxima. The next maximum point is decided as the maximum within the area which is not closer than the blocking constant $\beta = 4$ to the preceding maxima. Extracted maxima are: 1st = right-lower, 2nd = center, 3rd = right-middle, 4th = left. The results were refined hierarchically by the same procedures as shown in Figs. 11, 12, and 13. CPU time = 27.6 + 4 × 14.4 + 4 × 183 s.

polygon. By using directional information, the performance was increased and the extent of the variation of the sides could be expanded. As a result, the upper-left structure was extracted correctly in Fig. 12. Here, the amplitude of the Sobel edge operator was regulated by cutting off edge values lower than the average and by clipping values that were greater than the average plus standard deviation.

The improvement is clearer in the case of the second image shown in Figs. 13 and 14. In this case, the picture is darker, the contrast is lower than in the first image, and the center part is brighter because of

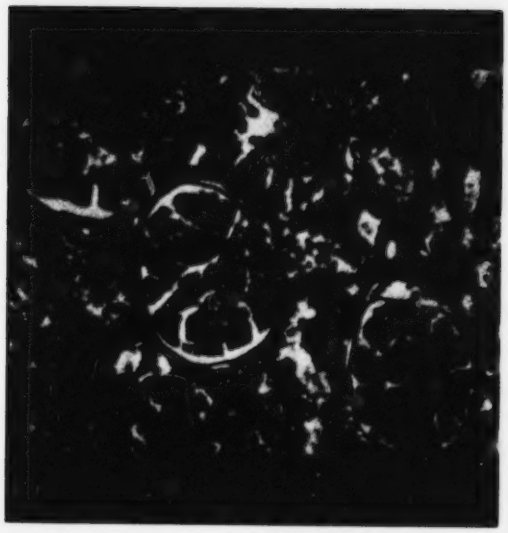


Figure 12. First four maxima using the directional information. By using the directional evaluation, the performance of the method is increased. The structure at the upper left had a variation of more than double in length at the right side. The DP matching was applied hierarchically. On the upper level of hierarchy for pitch $\alpha = 8$, the variation constant $\lambda = 3$, $\varepsilon = \infty$. On the lower level $\alpha = 1$, $\lambda = 1.2$, $\varepsilon = 16$. The correct shapes were extracted as the first three maxima. CPU time = 401 + 4 × 1016 s.

the lighting of the microscope and the shading of the camera. Hence, when we used gray level information only, the results were concentrated in the central part of the image, and the correct answers could not be obtained as shown in Fig. 13. By using directional information, the result was improved and five of the eight structures were extracted.

In the cases of Figs. 11 to 14, the problem is when to stop extracting the next shape. This requires verification by using other information, including man-machine interaction. At any rate, it is possible to measure quantitatively the width of the ring, the area of the structure, and so on.

3.5 Conclusions

In this chapter, the method of Kovalevsky (12) and Sakoe (13) was augmented by using, as in ref. 36, directionality for high performance and the

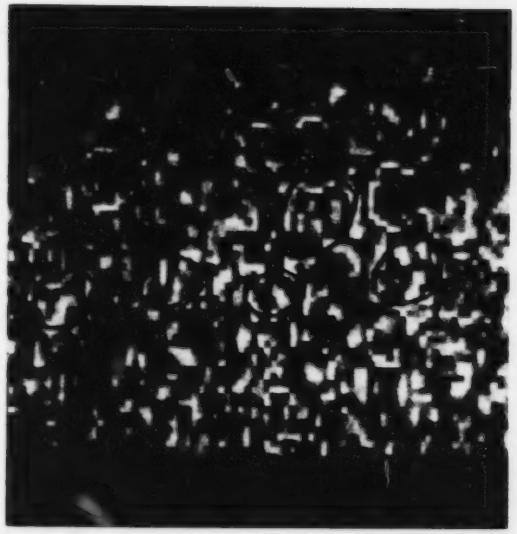


Figure 13. Sampled image with pitch $\alpha = 8$ and the first five maxima using gray level, $\lambda = 2$, $\rho = 32$, $\beta = 4$, array = 80 × 60, CPU time = 101 s.

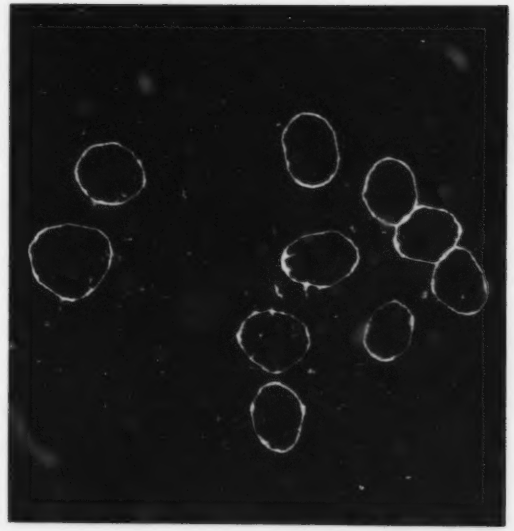
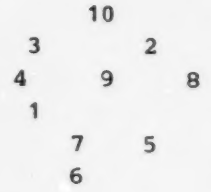


Figure 14. Original image with low microscopic magnification and first 10 maxima using direction. The method was applied hierarchically. As the rough extraction, $\lambda = 2$, $\delta = 1$, $\rho = 32$, on sampled image of $\alpha = 8$. Then, on original image of $\alpha = 1$, $\lambda = 1.2$, $\varepsilon = 16$. CPU time = 238 + 10 × 381 s. The order of the maxima was as follows:



hierarchy for reduction of calculation time. The cyclical structure of the contour was also considered. The method was applied to locate the glomeruli in microscopic images of kidney tissue sections. As can be seen, the proposed method is general, has a simple frame or scheme, and has a good ability to match objects to models.

The ultimate objective is to measure the area of each glomerulus and the area enclosed by the white ring. This objective is easier to realize through the DP recognition process.

Hereafter, while preserving the generality of the method, the individuality of each problem should be investigated to refine the method. In the case of kidney structure, there is some room for improving and adjusting the size of the edge operator. Also, some higher order features, such as pairs of parallel edges, could be investigated.

CHAPTER 4. RECOGNITION OF OCCLUDED, REFLECTIVE, AND TRANSPARENT OBJECTS WITH UNCONSTRAINED BACKGROUND AND ILLUMINATION BY USING INDUCTIVE DYNAMIC PROGRAMMING MATCHING

4.1. Introduction

There are two approaches for 3D scene analysis. One approach is to establish a correspondence between an input scene and a 3D model description directly in the 3D domain (3D approach) (38). In the other approach an input scene is considered as a 2D image, and the matching process is done on the 2D image (2D approach) (39).

In the 3D approach, a reference model is described completely. In the case of the generation of a view from the model as in computer graphics, this kind of framework is thought to be reasonable, because the transformation from the inner description to the view is straightforward. However, in the case of the processing of the input it becomes difficult, because we must search the whole transformation parameter space in the 3D domain. In fact, we could dare to say that there is no straightforward automatic method for the direct matching of a 3D model effectively.

Moreover, from the standpoint of a viewer (user), a direct 3D matching scheme is not necessarily required. To identify or locate an object, we do not need a complete description of a 3D model. For example, in the case of a circular disk, we can recognize it from the 2D view as a circle or an ellipse, except in the special case when the view axis and the disk are colinear. More extremely, we could say that an object like the disk can be recognized only by a planar circular model while allowing for a certain extent of variation. The tilt up to ±60° in the 3D domain changes the ratio of the long diameter vs. the short diameter from 1 to 0.5 in the 2D view (see Fig. 15). Conversely, if we allow 10% as the variance of the ratio of diameters in the 2D image, then the allowance for the tilt in the 3D domain becomes ±26°. Consequently, some extent of variation in the 3D world can be absorbed as a variation on the 2D image. After locating the object, we can verify whether or not the result is correct, and measure the 3D attitude. That is a rather easy job.

What must be stressed here is the flexibility of the matching method. When we pose the problem in this framework, then the ability to absorb variations is indispensable, even in the case of solid objects.

Flexibility is needed not only for absorption of the 3D attitude but also for the noise content of the input image. When objects are reflective and/or the background is textured, improper shapes or features will appear in the observed image. Sometimes the reflection or the shadow is stronger

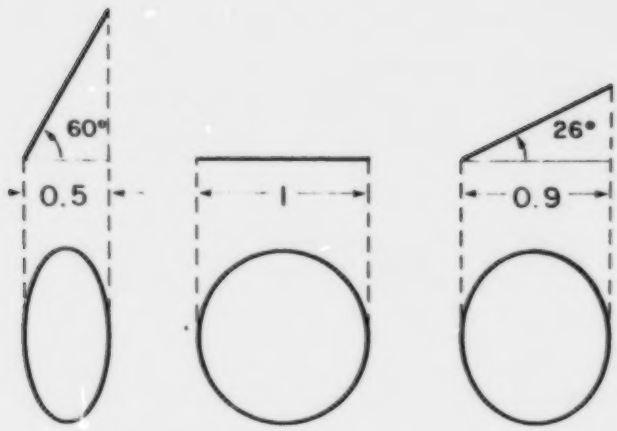


Figure 15. Variations in 2D domain caused by 3D tilt.

than the contour of the real object itself. When objects are occluded by each others, some features are missing (39-41). When objects are transparent, the feature does not come out where it would be because of the transparency and the diffraction. All of these effects may be viewed as some kind of "noise" in the wider sense when recognizing the object.

The image treated here is general not only in the sense that all the stated factors are present, but also that the background is gray (and textured in some case). The object is brighter than the background in some places and darker in some others. We cannot use a simple thresholding technique. All these factors make the matching problem difficult.

In this chapter, Inductive DP matching method is used to attack this problem. The input representation is point-like and the reference representation is line-like, as in refs. 12 and 13. However, in order to increase the performance, directional information is taken into account as in ref. 3. The reference model is a polygonal approximation of a contour. The cyclical property of the contour is used here.

4.2 Two-dimensional Dynamic Programming Matching Method

Figure 16 shows the block diagram of the process. An original image has 640 × 480 resolution and 256 gray levels (see Fig. 18). At first, it is sampled with the pitch α_1 by summing up the values in the $\alpha_1 \times \alpha_1$ area (see Fig. 16). Then, from the 3 × 3 Sobel edge operator, we get the edge amplitude a'(k, l) and the edge direction b'(k, l). Next, the sampling at pitch α_2

Sampling by $\alpha_1 \times \alpha_1$

↓

3×3 Sobel edge operator

↓

Sampling by $\alpha_2 \times \alpha_2$

↓

DP matching

Input : $\{a(i, j), b(i, j)\}$

↓ Reference model : $\{r_m\}$

Loop for next maximum

Figure 16. Block diagram of the process.

$$[48] \qquad a(i, j) = \max_{(j-1)\alpha_2 < l \leq j\alpha_2} \max_{(i-1)\alpha_2 < k \leq i\alpha_2} a'(k, l),$$

$$b(i, j) = b'(k, l)$$

where

$$[49] \qquad (k, l) = \arg_{(k,l)} \max_{(j-1)\alpha_2 < l \leq j\alpha_2} \max_{(i-1)\alpha_2 < k \leq i\alpha_2} a'(k, l)$$

and the amplitude is normalized between 0 to 1 after cutting off the mean value and clipping at the value of the mean plus the standard deviation.

Then, the DP matching method which is almost the same as the one used in Chapter 3 is applied. That is, an input representation is

$$[50] \qquad U = (a(i, j), b(i, j)), \quad 1 \leq i \leq I, \quad 1 \leq j \leq J,$$

$$0 \leq a \leq 1, \quad -\pi/2 \leq b \leq \pi/2$$

and the reference model, the shape of which we wish to extract from the input image, is

$$[51] \qquad R = (r_1, r_2, \ldots, r_m, \ldots, r_M)$$

where R is a sequence of M (= 16) line segments (vectors), the length of line segment r_m is $s_m = |r_m|$, and the direction is t_m. The global similarity is

$$[52] \qquad S(R, U) = \max_{\{(i_m, j_m), m=0 \sim M\}} \sum_{m=1 \sim M} d(i_{m-1}, j_{m-1}, i_m, j_m)$$

This is solved by using DP decomposition.

The DP has the following characteristics.

1. The reference model is a cycle of a contour which is expressed by piecewise linear segments.

2. The input image respresentation is pixel-wise.

3. The directional information is used to enforce the result.

4. The inductive DP matching is used.

5. The cyclic DP stated in Section 2.5.4 is used. The distance between the beginning point and the end point is evaluated.

6. The multi-resolutional DP stated in Sections 2.7.4 and 3.3.2 is applied. The number of calculations is reduced.

7. Constraint on the resultant positions. This makes the multi-resolutional scheme effective, and also makes the calculations decrease when there is a positional constraint.

8. Finding many objects in an image, as stated in Section 3.3.4.

4.3 Experiments with Scissors and Metal Rings

In the experiments, two groups of images are used, both of which are difficult to recognize. One contains reflective objects: three images of scissors and metal rings, named SR-0, SR-1, and SR-2 (see Figs. 17, 18, and 21); and the other has transparent objects, four images of ashtrays made of transparent glass, named AT-0, AT-1, AT-2, and AT-3 (see Figs. 25, 26, 30, and 31). All the images were taken through a TV camera and digitized into 640 × 480 pixels and 256 gray levels.

In the case of images SR-0, SR-1, and SR-2, two lights were used from both sides in addition to the normal room lights. The image called SR-0 was used to construct reference models of the shapes (see Fig. 17). The two rings on the left side in the image are metal fittings for a camera, one faces up and the other down. The lower left object is a filter for a camera, and semi-transparent. The center object is scissors. Four reference models of shapes were made: two for the metal ring and two for scissors. The scissors are partly reflective (metal) and partly black coated. These two objects were taken as the representatives for circular objects and generally shaped objects. Let us give names to each of the reference models: RL is the large circle of the metal ring, RS is the small circle of the metal ring, SL is the four-finger side of the scissors, and SS is

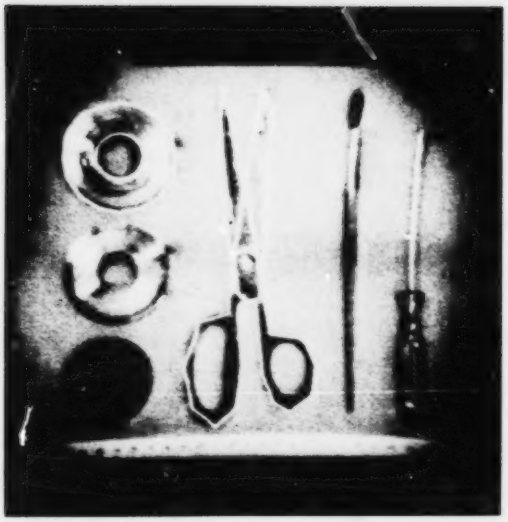


Figure 17. SR-0.

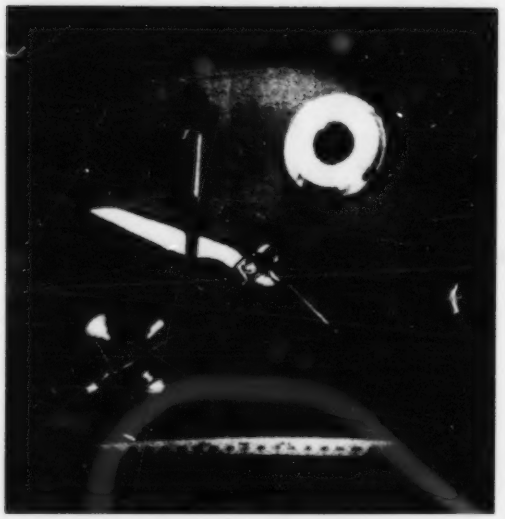


Figure 18. SR-1.

the thumb side of the scissors. In the cases of SL and SS, the position of the rotational center was also supplied and stored.

Figure 18 shows the image SR-1, and Fig. 19 shows the edge amplitude obtained by Sobel operator. The amplitude was normalized by cutting off at the mean value and clipping at the mean + standard deviation. Though the proposed matching method is hierarchical as explained in Section 3, all of the results are shown on 80 × 60 images, like the one in Fig. 20. The rough or principal performance of the system is almost decided by the first stage with the low resolution. The values of the constants in the experiments for scissors and metal rings are: first sampling $\alpha_1 = 1$, second sampling $\alpha_2 = 8$, length variation $\lambda = 1.2$, width variation $\omega = 1$, gap weight $\gamma = 1$.

Figure 20 shows the sampled image by compressing 8 × 8 pixels into one pixel: the two best extracted shapes for the large circle, and the best three for the small circle. The order of detection for the large circle was upper-right and then lower-left. The order for the small circle is upper-right, lower-right, and then lower-left. The processing time on VAX - 11/780 was 179 s for the small circle and 294 s for the large circle. The time to extract the subsequent maxima is negligible, so the total time to extract the shapes in Fig. 19 was 179 + 294 s.

Up to now, each result has been obtained separately. In order to connect these results, the distances between the centers of extracted large circles and those of the extracted small circles are measured as shown in

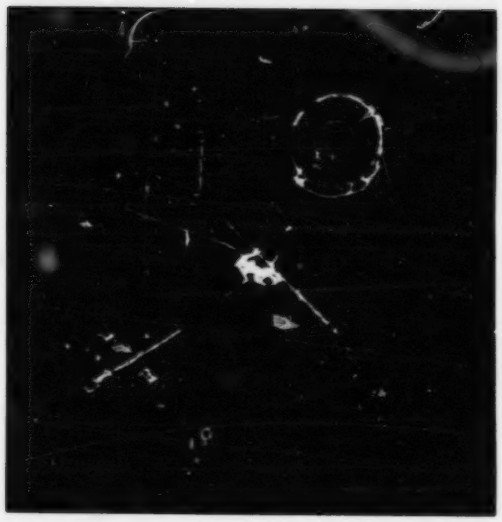


Figure 19. Edge amplitude, $\alpha_1 = 1$.

Figure 20. Rings on 80 × 60 image

Table 2. It shows that the first result for the small circle and the first result for the large circle turn out to be one object, and the third small and the second large circle turn out to be another object. As a result, both objects are extracted correctly.

The matching method is not rotation invariant. Hence, 36 references, each of which has different rotational angle, were generated automatically, and the matching was done with each of them. Figure 21 shows the extracted shape which had the best evaluation among the 36 references. The processing time was 445 s × 36 for the smaller (thumb) side of the scissors, and 463 s × 36 for the larger side. In this case, the resultant center of gravity was calculated as given in Table 3. In this case, both sides of the scissors were extracted correctly as the first evaluation,

Table 2. Center of gravity of extracted metal rings (SR-1)

Small circle (X,Y)	1	2	3	4	5
Large circle (X,Y)	(412, 100)	(377, 316)	(156, 356)	(116, 356)	(452, 92)
1 (416, 92)	8.9	227.5	370.5	399.6	36.0
2 (157, 359)	363.5	223.7	3.2	41.1	397.9
3 (279, 282)	225.0	103.4	143.8	179.2	256.6
4 (307, 171)	126.8	160.8	238.8	265.9	165.1
5 (416, 324)	224.0	39.8	261.5	301.2	234.9

though this matrix is also useful for verification of the result. Although the extracted positions for the tips of the scissors are not accurate, actually it is sufficient in many cases, to grasp the object, for example.

Figure 23 shows the edge amplitude from another test image SM-2 (see Fig. 22), where the background is the tiles on the floor. There are many texture edges in the background, which make the recognition difficult.

Figure 24 is the results of three maxima for the large circle and five maxima for the small circle. Table 4 shows the distances between the centers of gravity. From this table, we can point out the upper-right object as the fifth result for the small circle and the third for the large. However, the upper-left object was not extracted. This is because the shadow of the ring is very strong, while the edge of the ring itself is very weak, as can be seen in Fig. 23. Both sides of the scissors are extracted as the first results. However, we cannot verify this by using Table 5, because

Table 3. Rotational center of scissors (SR-1)

Small side (X, Y)	1	2
Large side (X, Y)	(316, 248)	(270, 271)
1 (310, 249)	6.0	46.2
2 (270, 271)	26.8	78.1

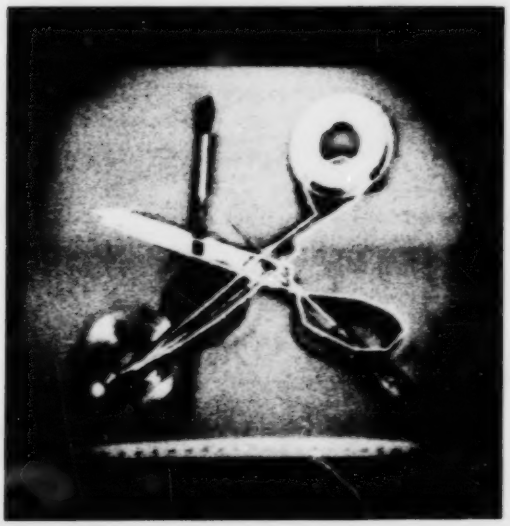


Figure 21. Scissors on SR-1.

Figure 22. SR-2.

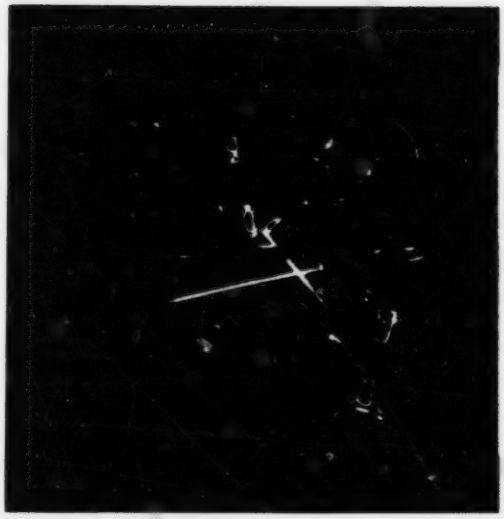


Figure 23. Edge amplitude.

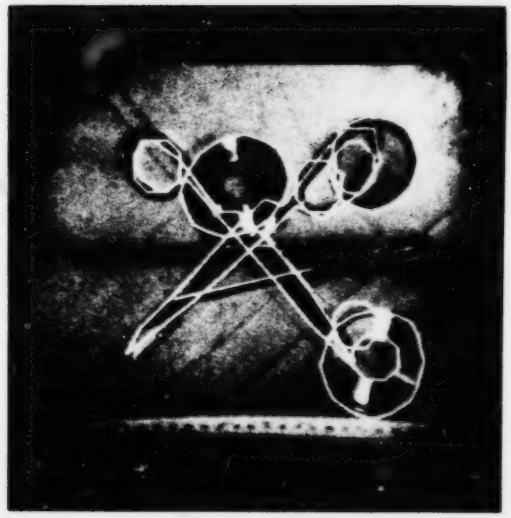


Figure 24. Results of SR-2.

Table 4. Center of gravity of extracted metal rings (SR-2)

Small circle (X, Y)	1	2	3	4	5
Large circle (X, Y)	(184, 124)	(432, 320)	(384, 148)	(428, 124)	(456, 360)
1 (452, 120)	268.0	201.0	74.0	24.3	240.0
2 (280, 148)	99.0	229.5	103.5	149.9	275.5
3 (453, 367)	362.5	51.5	229.8	244.3	7.6
4 (398, 220)	234.1	105.8	73.3	100.7	151.7
5 (376, 392)	329.7	91.2	244.1	273.0	86.2

Table 5. Rotational center of scissors (SR-2)

Small side (X, Y)	1	2	3
Large side (X, Y)	(297, 213)	(334, 254)	(359, 204)
1 (298, 205)	7.3	60.3	61.1
2 (331, 244)	45.7	10.7	48.9
3 (343, 263)	60.4	8.7	51.9

there are other couples which have short distances. In order to verify the result, we need further information from the original image.

4.4 Experiments with Transparent Ashtrays

In the case of the glass ashtrays, the usual fluorescent lamps on the ceiling were the only lighting source. Two reference models were made from AT-0 (see Fig. 25). Both have (M =) 16 sides or segments. The upper two objects are the ashtrays which we are using now. The upper-left one is right-side-up, and upper-right is upside-down. Lower-left is a side view of the ashtray. Lower-right is another ashtray which has more indentations than what we are using now.

Figure 26 shows the ashtray image AT-1 on a black background. Figure 27 shows the edge amplitude by 3 × 3 Soble operator after 2 × 2 sampling ($\alpha_1 = 2$). In the case of the ashtray images, the edges of the objects are not so clear. We found that the 3 × 3 edge operator after 2 × 2 sampling (effectively 6 × 6 edge operator) was better than the 3 × 3 edge operator on the original 640 × 480 image.

The original input image has so weak gray levels that histogram equalization (see Fig. 28) was used to improve the results. (However, in the case of AT-2 and AT-3, the results were worse.) Figure 29 shows the result by using histogram equalization before the sampling by $\alpha_1 = 2$, and then 3 × 3 edge operator, and 4 × 4 sampling ($\alpha_2 = 4$) (see Fig. 16). Other constants are $\lambda = 1.2$, $\omega = 2$, and $\gamma = 1$.

The circle is extracted by using the same method as in the case for the image of the metal rings. In the case of the square, it has rotational

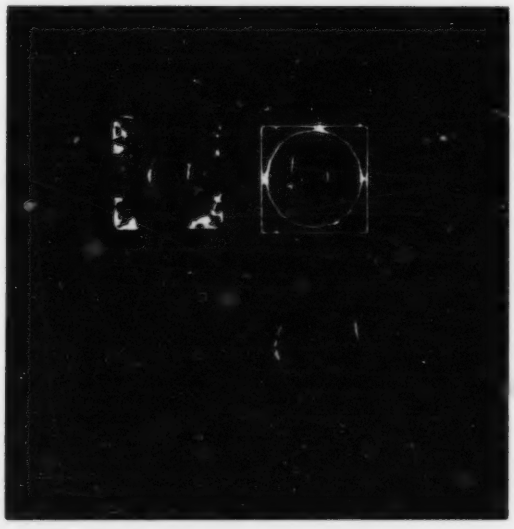


Figure 25. AT-0.

Figure 26. AT-1.

symmetry, so nine references, rotated by 0 to 80°, were used as the references. A total of 18 results, two maxima at each rotational degree are taken as candidates, and then they are sorted by the evaluation values. Figure 29 shows the results, five maxima for the circle and seven maxima for the square. In the case of the square, we did not block the results between different reference rotations. Consequently, there are several results for one object.

Figure 30 shows the image AT-2, ashtrays on documents, that is, one example of general background. Experimental parameters are no histogram equalization, $\alpha_1 = 2$, $\alpha_2 = 4$, $\lambda = 1.2$, $\omega = 2$, and $\gamma = 1$. The results are five maxima for the circle and six maxima for the square.

Figure 31 shows the image AT-3, where the ashtrays are on floor tiles as background. Experimental parameters for the circle are no histogram equalization, $\alpha_1 = 1$, $\alpha_2 = 8$, $\lambda = 1.2$, $\omega = 1$, and $\gamma = 1$. Parameters for the square are, no histogram equalization, $\alpha_1 = 2$, $\alpha_2 = 4$, $\lambda = 1.2$, $\omega = 2$, and $\gamma = 1$. The results are 5 maxima for the circle and 12 maxima for the square. Although there are many duplications and miss extractions, almost all shapes hit reasonable places.

In the experiments for the ashtrays, the parameters were different from each other. If we use the same conditions, the results become worse. This shows that these images are almost at the limit for quality for extracting objects by using the proposed method. To improve the results, we can not only adjust the parameters of the method or develop higher

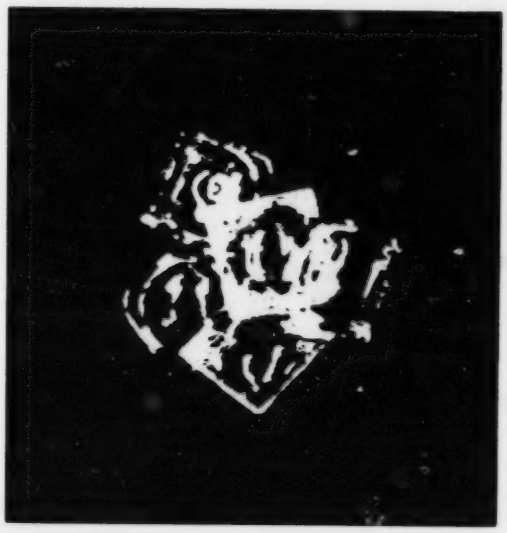


Figure 27. Edge amplitude, $\alpha_1 = 2$

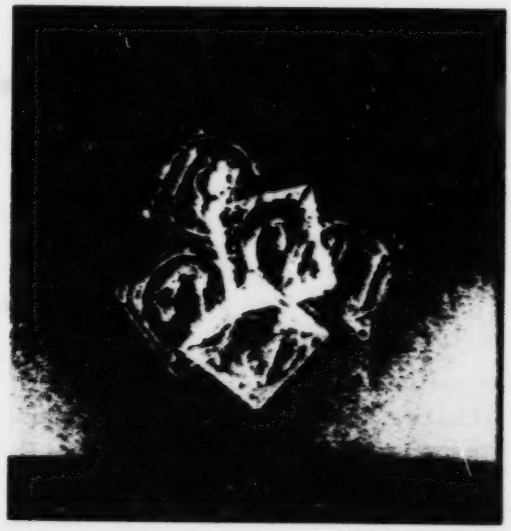


Figure 28. Histogram equalized AT-1.

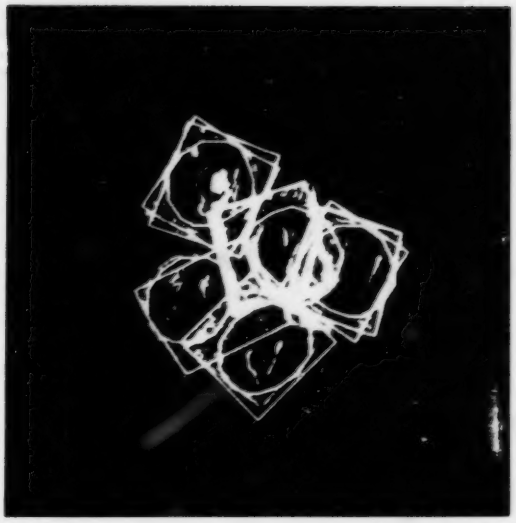


Figure 29. Results of AT-1.

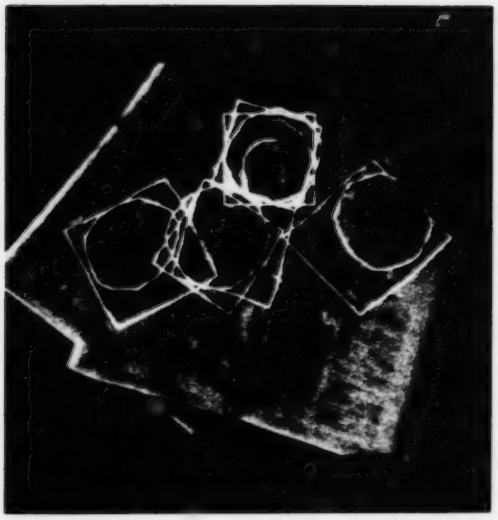


Figure 30. Results of AT-2.

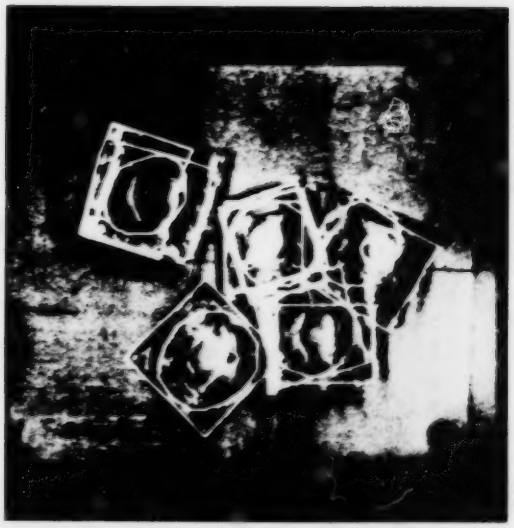


Figure 31. Results of AT-3.

performance method, but also rearrange the input environment, such as diffusive lighting condition, restricted background, or non-occlusive objects, and soon.

4.5 Discussion

In this chapter the method of Kovalevsky (12) and Sakoe (13) was augmented by using directionality for high performance and hierarchy for

reduction of calculation time, as in ref. 3. The cyclical structure of the contour was also considered. The method was applied to locate the reflective and/or transparent objects. Interestingly, the refrection and the transparency are also the subject of further investigations in computer graphics. As can be seen, the proposed method is general and straightforward, has a simple frame or scheme, and has a good ability to match objects to models.

In the area of computer vision, especially for robotics, the processing time is sometimes said to be critical. In this chapter, the processing time was not taken as the first constraint. Rather, an acceptable speed for the experiments was aimed at when a system was constructed. The reasons are:

1. We do not aim at a system based on the current state of hardware techniques, but would rather indicate a direction for future hardware architecture.

2. Once a higher performance system has been constructed, it is relatively easy to construct a deductive system to suit a given processing time, hardware, or lower performance.

3. Especially, DP has the property that the more constraints are given, the less computing time it takes. That is a different advantage compared with other analytical mathematical methods.

Hereafter, while preserving the generality of the method, the individuality of each problem should be investigated in many situations to refine the method.

CHAPTER 5. ROTATION-INVARIANT CONTOUR DYNAMIC PROGRAMMING MATCHING METHOD FOR THREE-DIMENSIONAL OBJECT RECOGNITION

5.1 Introduction

A major difficulty of recognizing 3D objects, compared to 2D object recognition, is due to the freedom of orientational attitude of a 3D object, i.e., rotations around z-axis (roll), y-axis (pitch), and x-axis (yaw) with respect to the camera coordinates. There are two approaches for solving this problem. One approach is to establish a correspondence between an input scene and a 3D model description directly in the 3D domain (the 3D approach) (38, 42). In the other approach, an input scene is considered as a 2D image and the matching is done on the 2D image (the 2D approach) (39).

In the 3D approach the reference model is described completely and the orientational variations of the object in an image are to be found analytically. Given the rotations, the generation of a "view" from a computer graphics description is straightforward. However, the processing of the input image is an inverse operation where, at best, only a partial description of an object may be found. In the process of comparing the models to the objects, many possibilities have to be considered.

The 2D approach basically uses some form of template matching. To identify or locate the objects with the 2D approach, we do not need the complete description, i.e., the 3D model. After locating the object, we can verify whether or not the result is correct and measure the 3D attitude. However, in the case of rigid templates the number of templates required to match all the possible 3D views becomes very large, even though some of the variations in the input can be absorbed as tolerances.

When we pose the problem in the 2D framework, then the flexibility of the matching method and the ability to absorb variations are indispensable. Flexibility is needed not only to absorb the perturbations in the 3D attitude but also for the noise content of the input image. In this chapter, a DP matching method is proposed which is invariant to the rotation around the view-axis (z-axis) of the camera, while the rotational variations around the other axes are represented by many different "views" as reference models. The flexibility of the represented DP method minimizes the required number of reference models.

The DP matching method proposed by Yamada (3) for handprinted Chinese character recognition had the following properties:

1. Use of linearly approximated directional segments for both input and model contours. This reduced the calculation compared to pixel-wise

representation of input, as in refs. 12 or 36, while retaining the performance.

2. One-to-one, one-to-two, and two-to-one correspondences between model and input segments, allowing variations of segment lengths.

3. Introduction of the "precedable segments," i.e., local inter-dependency (local ordering) among input features. As a result, a break in a line (change of topology) was allowed, while retaining the high calculation speed.

4. Global evaluations are possible as the hierarchical DP minimum spanning tree is used for evaluating between contours.

The major weakness of the existing DP methods, when we want to apply them to 3D object recognition, is that they are rotation dependent. In this chapter, the method in ref. 3 is simplified and augmented to make it rotation invariant. In order to make the system simple, the properties numbered 3 and 4 are not included. Instead, rotation invariance and cyclicity of model contours are introduced. The properties of the proposed method are summarized below (43).

1. Use of edge segments. The input segments are linear approximations of the edge. The model segments are longer and linear, but can be, conceptually, curved approximation of the edge.

2. One-to-many (model-input) correspondence instead of two-to-one (model-input) correspondence as in ref. 3. This makes the structure of the DP calculation simple (but the calculation increases). This framework is reasonable, because the input is a linear approximation and the model is a cursive approximation. It can be assumed that one input segment does not correspond to multiple model segments.

3. Rotation-free evaluation by using the relative angle between two succeeding segments.

4. Cyclic matching as used in a different type of DP method (36).

5.2 Input and Model Representation and Global Distance

The input scene is represented by line approximations of the edges. At first, the gradient and the surface normal are calculated by using 3 × 3 information. The next stage consists of grouping or clustering the pixel features to segment the image into surfaces (facets). This consists of three steps: first, to find a mapping which transforms selected localized

parameters into a new space where the desired surface types form clusters, second, to detect and mark the clusters, and third, to use the cluster information to locate the surfaces. Then, the edges, which are the region boundaries, are approximated by linear segments. The details are described in ref. 42.

The edges of the unknown input U are approximated by linear line segments (continuous piece-wise linear contours)

$$[53] \qquad U = \{u_i\} = \{u[i]\}, \qquad 1 \leq i \leq I$$

Each segment is directed so that the left side is a valley and the right side is a hill. If the direction was ambiguous, two segments which have opposite directions were tabulated separately. Consequently, each segment has a unique direction. In Fig. 32, u[i'] is a "preceding" segment of $u[i_b]$, i.e., the tail of the arrow $u[i_b]$ is touching the head of u[i']. Sometimes this relation is expressed by $i' = i_b - 1$ or $i_b = i' + 1$.

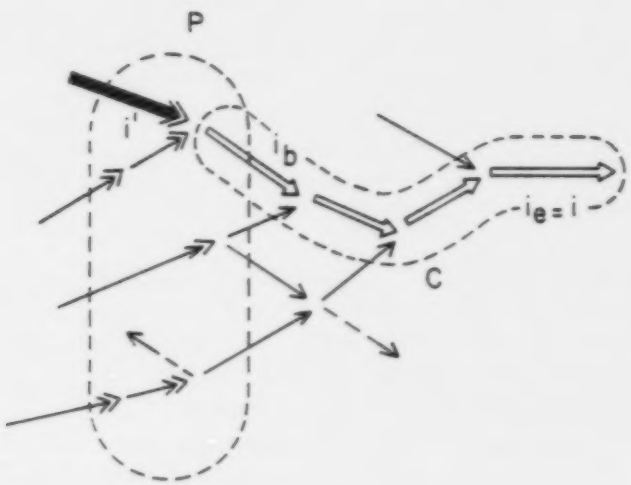


Figure 32. Travelling in input segment tree. Double arrowed segments are in precedable set P. Set P is defined as the corresponding input segments to r[m - 1] when the correspondence of r[m] and u[i] is assume d. When u[i'] is fixed among the elements in P, corresponding elements of C are defined. The dashed arrow which has the opposite direction to the travel has no relation to the present calculation.

The attributes of each segment are length v[i], direction w[i] from the beginning point (tail) to the end point (head), and the coordinate $(x_{u[i]}, y_{u[i]})$ at the end point. The beginning point of u[i] is identical to the end point of the preceding segment of u[i].

$$v[i] = ((x_{u[i]} - x_{u[i-1]})2 + (y_{u[i]} - y_{u[i-1]})^2)^{1/2}$$
$$w[i] = \arctan\left\{(y_{u[i]} - y_{u[i-1]})/(x_{u[i]} - x_{u[i-1]})\right\} \quad [54]$$

When segment u[i'] and u[i] are "connected", see C in Fig. 32, where C is the collection (series) of segments, the attributes are defined as

$$V[i', i] = \sum_{i''=i'+1}^{i} v[i''] \quad [55]$$

$$W[i', i] = \arctan\left\{(y_{u[i]} - y_{u[i']})/(x_{u[i]} - x_{u[i']})\right\}$$

Note that here the segment u[i'] is excluded from the calculation of V[i', i].

The reference model, the shape of which we wish to extract from the input image, is composed of a series of segments. In the experiment, the series forms a closed loop.

$$R = \{r_m\} = \{r[m]\}, \quad 1 \leq m \leq M \quad [56]$$

The attributes of each segment are length s[m], direction t[m], and coordinate $(x_{r[m]}, y_{r[m]})$ as in the input. It is assumed that one model segment corresponds to one or more input segments.

The objective here is to find the combination which gives the optimal correspondence between the model segment and the input segment(s) which minimize a global distance function D(R, U).

A set of input segments to which the model segment r[m] corresponds is expressed as (see Fig. 33),

$$C[m] = (u[i_b], u[i_b + 1], \ldots, u[i_e - 1], u[i_e]) \quad [57]$$

A local distance is calculated from the length difference (plus an input nonlinearity) d and the relative angle difference e. When r[m - 1] corresponds to C[m - 1] = (i" + 1, ..., i') and r[m] corresponds to C[m] = (i' + 1, ..., i), d and e are defined by

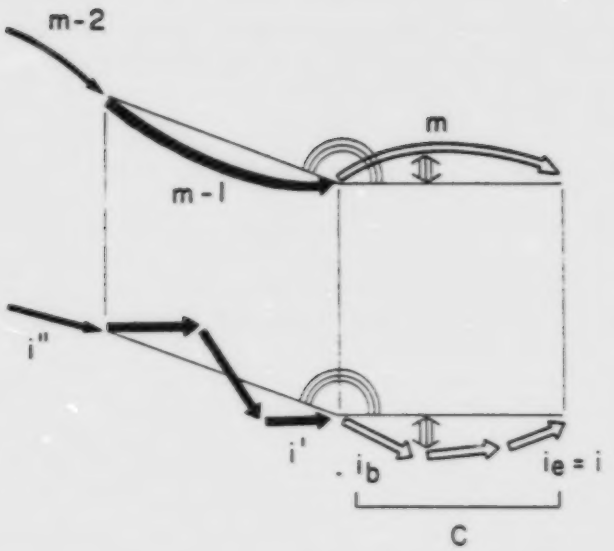


Figure 33. The rotational independent DP process. When r[m] corresponds to C (white arrows), the evaluation is calculated from (1) the difference of the "shape" between r[m] and C, and (2) the difference of the relative angles between <r[m - 1] to r[m]> and <u[i" to i'] to u[i' to i]>.

$$d(m, i', i) = \left|\left\{(x_{u[i]} - x_{u[i']})^2 + (y_{u[i]} - y_{u[i']})^2\right\} - s[m]\right|$$

$$+ \left|V[i', i] - \left\{(x_{u[i]} - x_{u[i']})^2 + (y_{u[i]} - y_{u[i']})^2\right\}^{1/2}\right| \quad [58]$$

$$e(m - 1, m, i", i', i) = \left|(W[i', i] - W[i", i']) - (t[m] - t[m - 1])\right|$$

The global evaluation value is obtained from the sum

$$E(\{C[m]\}) = \sum_{m=1}^{M} \left\{\delta \cdot e(m - 1, m, i"[m], i'[m], i[m]) + d(m, i'[m], i[m])\right\} \quad [59]$$

where $e(0, 1, i", i', i) = 0$, $\delta = 0.5$ (mesh/deg).

The optimum is expressed as the minimum of this value, i.e.,

$$D(R, U) = \min_{\{C[m] \mid m=1\sim M\}} E(\{C[m]\}) \quad [60]$$

5.3. Rotation-free Dynamic Programming Matching Method

In order to minimize the value $E(\{C[m]\})$, we use the cyclic DP technique. As an initial value for $m = M'(= [M/2])$, we set

[61] $$f(M', i) = 0, \quad h(M', i) = i - 1$$

For $m = M' + 1, M' + 2, \ldots, M - 1, M, 1, 2, \ldots, M$; we calculate the recurrence value $f(m, i)$ with respect to each input segment $u[i]$ as,

[62] $$f(m, i) = \min_{i' \in P} \{f(m - 1, i') + \delta \cdot e(m - 1, m, h(m - 1, i'), i', i) + d(m, i', i)\}$$

[63] $$h(m, i) = \arg \min_{i' \in P} \{f(m - 1, i') + \delta \cdot e(m - 1, m, h(m - 1, i'), i', i) + d(m, i', i)\}$$

where P is a set of admissible preceding segments which correspond to the end point of $r[m - 1]$, when the end point of the segment $r[m]$ is fixed to the end point of segment $u[i]$. P and its path C in Fig. 33 are defined by a tree search process by counting up all possible paths from $u[i]$ (44-46), (see Fig. 32),

[64] $$P = \{i' \mid s[m]/\lambda \leq V[i', i] \leq \lambda \cdot s[m]\}$$

Hence, $f(m, i)$ is the optimal evaluation with respect to $\{r[m'] \mid m' = 1, 2, \ldots, m\}$, where the length of each segment is varied within the range from $1/\lambda$ to λ under the assumption that the end point of $r[m]$ if fixed to $u[i]$. Here, $h(m, i)$ is a matrix for storing the correspondence. The principle of optimality, as expressed by Bellman (1), states that "an optimal policy has the property that whatever the initial state and initial decision are, the remaining decision must constitute an optimal policy with regard to the state resulting from the first decision." The meaning of this statement in the recurrence equation is "the optimal value $f(m, i)$ can be decided only by the local distance $\delta \cdot e(m - 1, m, i'', i', i) + d(m, i, j)$ with respect to m and the optimal value $f(m - 1, i')$ up to $m - 1$, whatever the path from 1 to $m - 1$ is."

After calculating the recurrence relation with respect to $m = 1, 2, \ldots, M$, the global distance D is obtained from

[65] $$D(R, U) = \min_{i} [\, f(M, i) - f_0(B(i)) + \gamma \cdot \{(x_{u[B(i)]} - x_{u[i]})^2 + (y_{u[B(i)]} - j)^2\}^{1/2}]$$

Note here that the obtained solution is optimal, that is, it is the same as the D(R, U) of Eq. [60].

5.4 Conclusions

In this chapter the method of Yamada (3) was simplified by using the continuity condition, one-to-multiple correspondence in the DP recurrence formula, and the method was augmented by using relative angles for rotational invariance. The cyclical structure of the model contour was also considered.

The proposed method is based on the 2D attributes (z = 0 always) of feature elements and is invariant to rotation around the z-axis (roll). In the present formulation the method requires a sufficient number of models to adequately represent the views created by rotations around x-axis (yaw) and y-axis (pitch). By using the 3D attributes from the range camera (x-, y-, and z-values of the contours), the method becomes invariant to pitch and yaw, as far as the segment corresponding to the model can be "seen" in the input scene. The number of models is now significantly reduced. A complete 3D model description requires a set of 2D models corresponding to the number of significantly different views of the object.

The method is a fundamental scheme for applying DP to the orientation-invariant matching. The method is general, simple, straightforward, and can be improved on the following points:

1. Use of m - 2 and m - 1 to calculate the recurrence function of m instead of only using m - 1, in order to fix all degrees of freedom (curvature and torsion) in the 3D space.

2. Introduction of discontinuous DP for scenes of low quality or occluded objects.

3. Introduction of the hierarchical DP scheme (3) for more global evaluation.

CHAPTER 6. TRANSPARENT OBJECT EXTRACTION FROM REGULAR TEXTURED BACKGROUNDS BY USING BINARY PARALLEL OPERATIONS

6.1. Introduction

The perception of transparency by the human observer is a very complex process. Quoting Metelli (47), "perceptual transparency depends on the spatial and intensity relations of light reflected from a relatively wide field and not on light reflected only from a local area." In the area of computer graphics, the transparency of an object has been investigated in order to increase the reality of the display of the object (48). Compared with this, in the area of image processing and recognition, there is a trend to avoid this type of complex environment. For example, in the application of "shape from shading," an opaque material with isotropic surface microstructure is assumed (49). The reason is that the processing of the input image is far more difficult than calculations for display, where the problem is a conversion from an internal description to an iconic image representation. Output processing (display) is complicated but conceptually straightforward, while input processing (image segmentation and recognition) needs elabolate analysis of the input image.

Here, by focusing on an extreme example, i.e., an "invisible" (transparent) object, we study the effects of adjusting the observing environment to extend the framework of input processing. In this chapter we do not intend to "recognize" such kinds of objects directly, but rather put the stress on creating some environments for such objects to see what can be achieved by rather restricted processing with commercially available machines which have binary parallel image operations.

6.2 Lighting for Transparent Objects

In the case of a transparent object, the observed image has intrinsically very little gray level difference between the object and the background. First, we investigate ways of controlling the lighting.

1. Usual lighting on uniform background. Figures 34 to 36 are ashtrays on black, gray, and white uniform backgrounds, respectively. In the figures the background levels do not seem to be very different, since the display was adjusted to create a good range. If the display was not regulated, we could not see the object well, because the difference of the gray levels was very small. However, if we compare the three images carefully, we can notice the differences between the shadow or reflection regions. In the case of the black background, the difference between highlight and background is large, because the brightness of the reflecting part is the same within all three images.

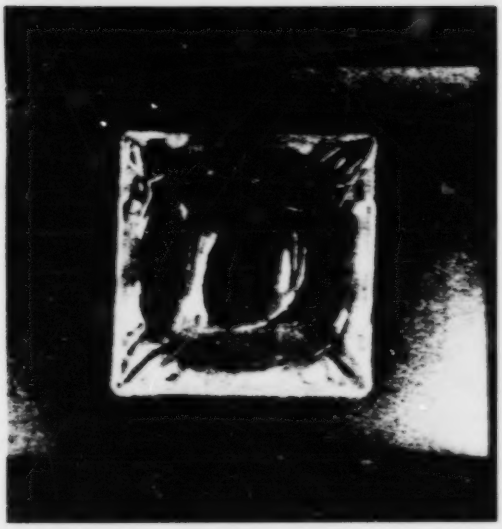


Figure 34. Image of the ashtray with lighting from top and black background.

Figure 35. Gray background.

When we pay attention to the shadow, we can see that it is stronger in the white background. At any rate, we cannot find a threshold which divides the image into the object and the background.

2. Polarized lighting and polarized filter. In ref. 50, the highlight of the reflection was used, but the study was restricted to cylindrical workpieces. In the general case, specular reflections should be eliminated as much as possible. One way to reduce the reflections is to use diffused lighting. Actually we were able to reduce the specular reflection fairly well. However, by reducing the reflection the signal from the background also becomes low. So, when the signal is amplified to create a good display, like in Fig. 37, the reflection is also enlarged. As a result, the effect of reducing the reflection is cancelled.

3. Diffused lighting. Diffused lighting is another way of reducing specular reflections (see Fig. 38). We can see that Fig. 38 is soft compared with Figs. 34-36. This is because the specular reflection is removed by the diffused lighting. However, the situation that the image must be regulated is the same as the case of the polarized lighting. One important conclusion from these experiments is that the more idealize the environment, the less visible is the object!

4. Diffused lighting from the bottom. In the case of transparent objects, we can also use the lighting from the bottom of the object. In ref. 51, this lighting system was used for printed wiring board inspection. Figure 39 is an image with diffused lighting from the bottom. This way of observation eliminates the reflection completely and catches the refraction (lens) effects of the object.

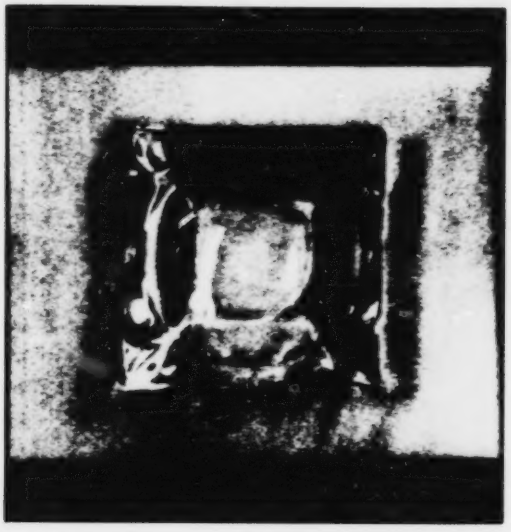


Figure 36. White background.

Figure 37. Polarized lighting.

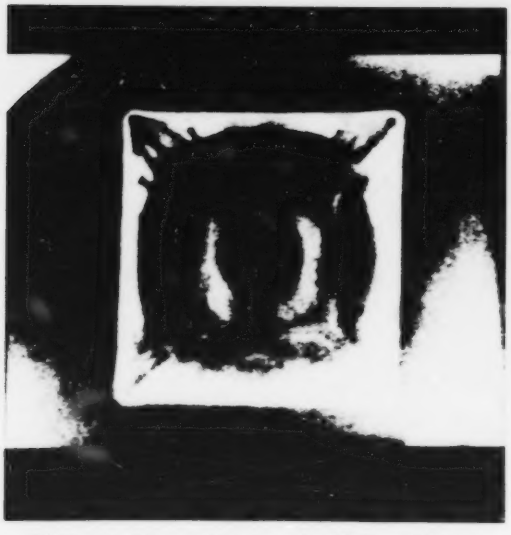


Figure 38. Image of the ashtray with diffused lighting from top.

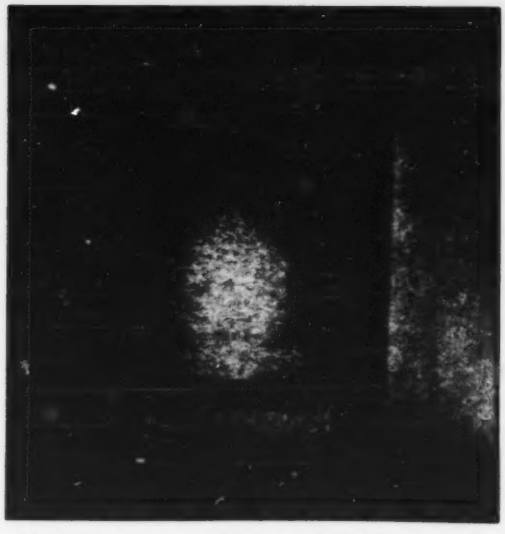


Figure 39. Diffused lighting from bottom.

6.3 Textured Background for a Transparent Object

Ideally, a transparent object will be invisible in some sense. Nevertheless, we can see the object due to several optical effects.

1. Reflection from the object. Usually a transparent object is also reflective and this information becomes important for detecting the object. In the cases of uniform background, see Fig. 35, the reflection gives many cues to detect the transparent object.

2. Absorption or translucency of the object. This gives the difference between gray levels of the object and the background. This phenomenon has been investigated mainly from the viewpoint of how the change of the gray level or color affects the human perception of transparency (47, 52). However, it seems almost impossible to detect this difference in the case of the ashtray in this experiment.

3. Warping by refraction (lens effect). Nonuniform thickness creates convex or concave lens effects and changes the image which is seen through the transparent material. Diffused lighting from the bottom brings out the refraction effect well as a change of gray level (see Figs. 39 and 40). The graph on the top of Fig. 40 shows the intensity changes which the observer can see through the transparent object or directly from the background. The darker parts in Fig. 39 are caused by the magnification effect of the edges of the ashtray which form a convex lens. The refraction also changes the shape of the background which can be seen through the transparent material. Sometimes the image of the background is out of focus.

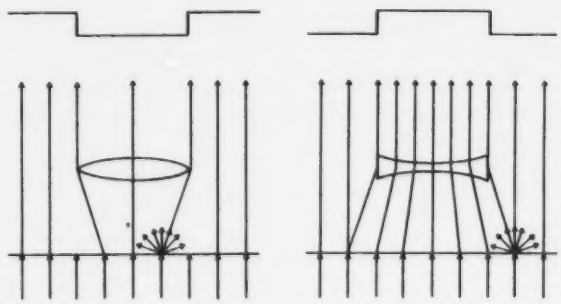


Figure 40. Variation of diffused light from bottom, (a) for a convex lens and (b) for a concave lens.

4. Shadow by refraction. Another important effect of refraction is a shadow on the background. The refraction gives some shadow on the background even if there is no absorption. Figure 41 shows this effect in the case of (a) a convex lens and (b) a concave lens. The rays from one far-away light source (the solid lines from the top) are returned to a far-away observer (the broken lines), as shown in Fig. 41. The graph on the bottom of Fig. 41 shows the brightness on the background, and the graph on the top shows the intensities which the observer can see. With many light sources these effects are superimposed.

5. Modification of medium. The visibility of a transparent object can be enhanced by placing it in a medium for which the difference between the refractive indices of medium and object is greater than that of air.

6. Observations with polarized light. Usually transparent objects have nonuniformities in microstructure or stresses and strains caused by the manufacturing processes which change the index of refraction. These irregularities can be observed by using defused polalized light from the back of the object and a polalized lens (camera).

7. Color selectivity. A medium has different refractive indexes for different light wave lengths. Hence, usage of different wave lengths (color) is sometimes useful.

We have described transparency from a physical point of view. In other words, in these cases the background is assumed to be more or less

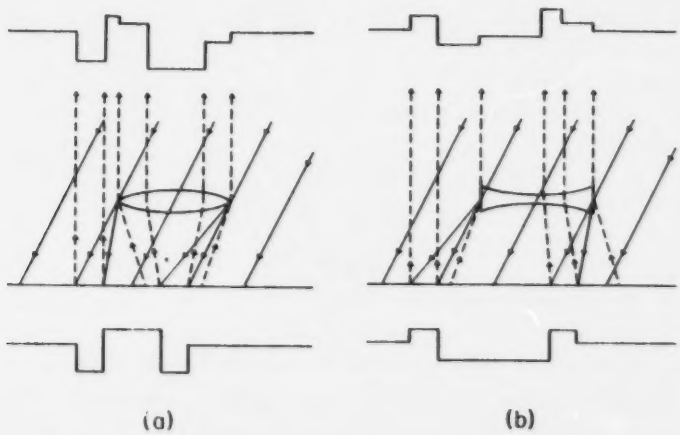


Figure 41. Shadow on the background (graph below) and the observed intensity change (graph above) by (a) convex lens and (b) concave lens.

uniform. However, for human vison the shape information gives a stronger feeling of transparency or cues for detecting a transparent object than the gray level (or color) change (52).

The first four effects can been seen in the normal environment. The remaining ones require special instrumentation. In this chapter, we concentrate on the aspects reflection, warping of the background shape, and shadow. We introduce a textured background to assist the automatic detection of transparent objects. If the background is textured, the refraction of the object may change the texture in some regions of the image. However, general irregular textures make automatic processing very difficult, hence we use a controlled regularly textured background. The idea is to remove the prepared regular texture. The remainder must then represent at least some parts of the object. For the processing, we use binary parallel operations which are available in recent commercial image processors.

6.4 Dilatation-Erosion Operations on Binary Images

The dilatation-erosion operation is a series of logical neighborhood operations which extracts small dots or thin lines while preserving large areas almost intact (53). This operation was applied to detect defects in printed circuits (54), and has been used as a fundamental operation in the course of developing dedicated image processors (55, 56). The dilatation-erosion operation is defined as follows.

(B1) b = BIN[s](a) : Binarization. At first, we binarize the input image a with the threshold s. The white part of the figure is logical 1, and the black part of the figure is logical 0 in the following logical operations.

(B2) LAR[n, t] : n-connected dilatation (expansion or making larger), t times.

$$[66] \qquad LAR[n, 1](f) = f(i, j) \text{ or } \left\{ \underset{(i',j')}{\text{OR}} \; f(i', j') \right\}$$

where (i', j') is n-adjacent to (i, j).

If the pixel itself or one of the n-neighbors of the pixel is one, then the result becomes one; otherwise the result is zero. The number n is either 4 or 8 for a square grid or 6 for hexagonal. This makes the white parts larger, and the operation can be done in parallel with respect to each pixel. This type of operation is most suitable for conventional parallel image processors and is usually performed within 1/30 s. This operation may be concatenated as

[67] LAR[n, 2](f) = LAR[n, 1]LAR[n, 1](f)

When t = 0, the operation LAR[n, 0] does not change anything.

(B3) SMA[n, t] : n-connected erosion (contraction or making smaller), t times.

[68] SMA[n, 1](f) = f(i, j) and $\{ \underset{(i',j')}{\text{AND}} f(i', j') \}$

where (i', j') is n-adjacent to (i, j).

[69] SMA[n, 2](f) = SMA[n, 1]SMA[n, 1](f)

If the pixel itself and all of the n-neighbors of the pixel are one, then the result becomes one; otherwise the result is zero. This is the opposite operation to LAR. This operation has the same property as LAR with respect to parallelism.

(B4) Dilatation followed by erosion operation.

[70] LGSM[n, t] = SMA[n, t]LAR[n, t]

By using this series of operations, small black (logical 0) areas whose diameters or widths are less than 2t disappear, while the larger areas remain almost as they were. This operation was used effectively and successfully for detecting defects in print circuits (54).

(B5) Erosion followed by dilatation operation.

[71] SMLG[n, t] = LAR[n, t]SMA[n, t]

This series of operations makes small white areas vanish since it is the converse of the former LGSM operation.

Conventionally, only the erosion-dilatation or dilatation-erosion operations are used as a sequence. In this chapter, we use the dilatation-erosion-erosion-dilatation operations and the dilatation-erosion-erosion-dilatation operations as two symmetrical sequences because of the regularly textured background.

6.5 Experiments

A dotted sheet is placed underneath a thick transparent glass plate and a transparent ashtray is placed on top of the glass, as shown in

Fig. 42. In Fig. 43, the background is a chessboard sheet similar to those often used to display transparent objects in computer graphics. However, in this case, the dots in the sheet are smaller, and the sheet is placed diagonally with respect to the camera.

In this chapter several words are used for indicating a texture element (texel). The word "dot" specifies a texel which has the same size as the original texel on the background. The word "spot" refers to a texel that appears larger than the original "dot." The word "point" specifies a texel smaller than the original "dot." The variation in the size of the texel is caused by refraction (lens effect). The word "region" is used for an area in the image which is considerably larger than the texel. However, it is not always possible to distinguish between a very small region caused by local specular reflection and/or noise.

Usually one "dot" consists of more than several connected pixels. If the size of the pixel is close to the size of the texel, then Moire patterns can occur, the digitization of the texels generates too many pixels of varying gray levels per texel, and gray level analysis of the texture becomes necessary. This problem can be alleviated by using two compensated thresholds, one for the black and the other for the white thresholded images.

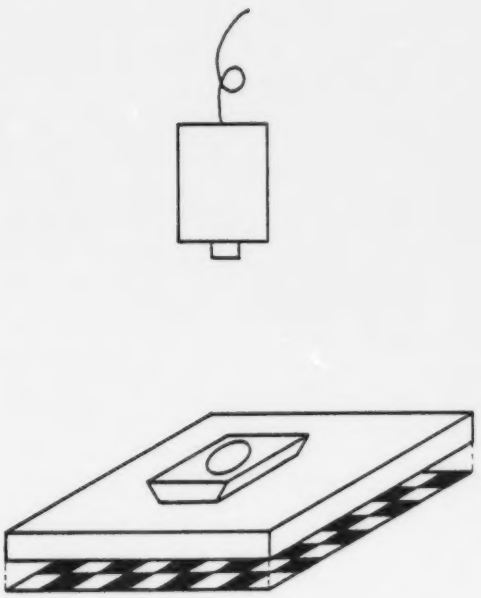


Figure 42. Schematic layout of the experiment with an ashtray on textured background.

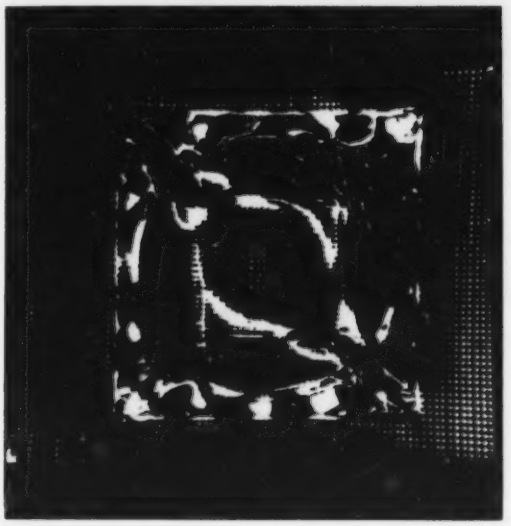


Figure 43. Gray level image a.

The picture is taken from the top by using a TV camera. (In the cases of Figs. 34 to 39, the thick flat glass was not used). The process is as follows.

(P1) b = BIN[s](a) : Binarization of image a with threshold s. The input image a has 256 gray levels and 640 × 480 pixels. In the experiments, the threshold s was selected by the operator for each image. However, as will be explained later, we need not care too much about the selection of the threshold. The following operations are done for the input which has a chessboard background (see Figs. 43 and 44).

(P2-1) c = SMLG[4, 1](b) : Preprocessing for small white point cleaning. By SMA operation, the white areas are made smaller. Small white points whose diameters are less than 2 disappear. Followed by the LAR operation, the large white areas are recovered almost intact. The result is seen in Fig. 45.

(P2-2) d = LGSM[4,5](c) : Background black dot (texel) removal. Operation LAR removes background black dots if the diameter of the dot is less than (2t =) 10. By the SMA operation, large black regions are recovered almost intact. This series of operations LGSM removes background black dots, while the large black regions remain, see Fig. 46. By concatenating SMLG[4, 1] and LGSM[4, 5] as

[72] LGSM[n, t]SMLG[n, t'] = SMA[n, t]LAR[n, t + t']SMA[n, t']

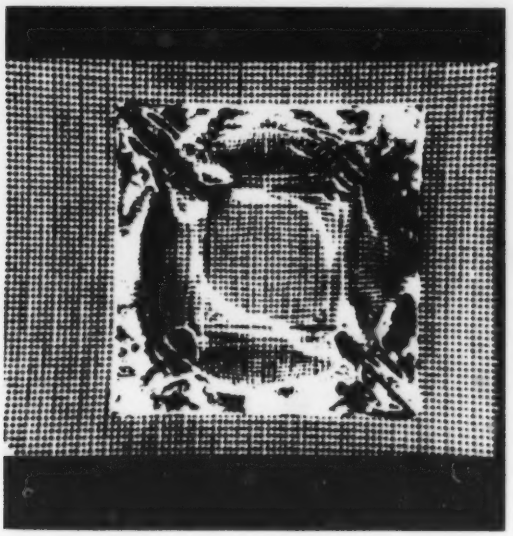


Figure 44. b = BIN[85](a).

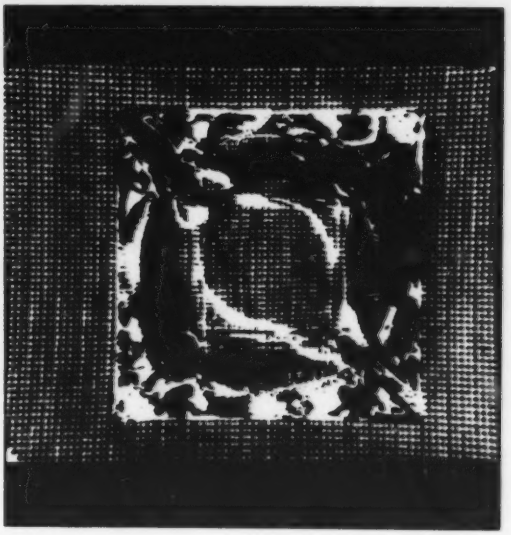


Figure 45. c = SMLG[4, 1].

small white points are converted into larger black regions and then normal black dots (texels) are removed. The small black points do notaffect to the result because the black points are removed at the stage of the black dot removal.

(P3-1) e = LGSM[4, 1](b) : Preprocessing for small black point cleaning. The same action as (P2-1) is carried out with respect to small black points.

(P3-2) f = SMLG[4, 5](e) : Background white dot removal. Background white dots disappear if the diameter of the dot is less than (2t =) 10. This series of operations changes the background to black regions, while the large white regions remain (see Fig. 47). The process (P3-1) and (P3-2) is exactly opposite to (P2-1) and (P2-2) with respect to LAR and SMA operations, and (P2) and (P3) are processed in parallel, that is, (P2) and (P3) are symmetric.

(P4) g = OR(NOT(d), f) : Unification of d and f. In (P2) and (P3), sequences of dilatation-erosion and erosion-dilatation operations were used symmetrically. By NOT(d), only the large black regions come out as logical 1. The operation OR unifies the large black areas and the large white areas (see Fig. 48). Intrinsically, this image (Fig. 48) is the final result of this kind of processing. However, in order to create more homogeneous regions, we do some postprocessing.

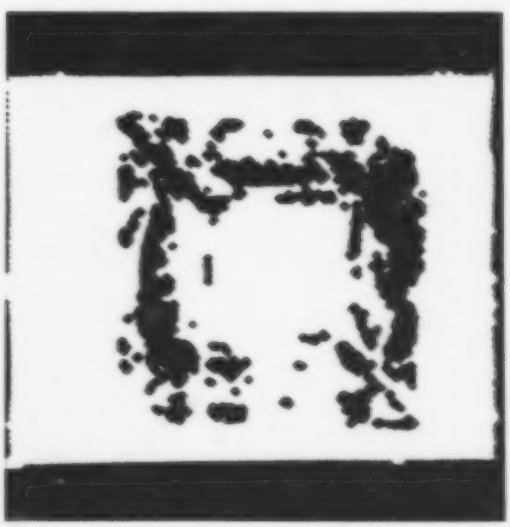


Figure 46. d = LGSM[4, 5]SMLG[4, 1](b).

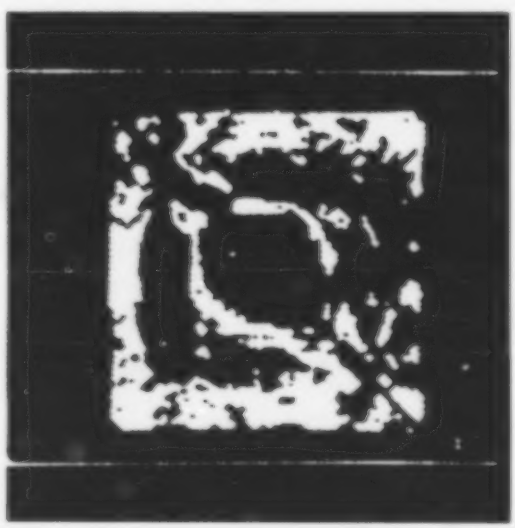


Figure 47. f = SMLG[4, 5]LGSM[4, 1](b).

(P5) h = SMLG[8, 10]LGSM[8, 10](g) : Postprocessing for small region removal. By LGSM operation, small black regions in the ashtray are removed. By SMLG operation, small white regions in the ashtray and the background are removed (see Fig. 49). In this case, we did not use any shape information for the object, although, we can say that the object is now almost "located." If we use some commercial image processor the whole process will be performed within several hundred milli-seconds, although it took 971 s (pre-processing = 86 s, main-processing = 300 s, post-processing = 585 s) by the VAX - 11/780 Fortran program.

In principle, in the case of an ideal black and white dot background, we can take any value as a threshold for binarization, because only the textural information is used in the processing and the texture is extreme black and white. Consequently, this is a threshold insensitive object detection method. During the experiments, the results did not change much when we changed the threshold.

Figures 50 and 51 show another case of square dot background. In this case, the shape of the black and that of the white texels are not the same. Hence, different constants were used for LGSM and SMLG operations, i.e.,

[73] g = OR {NOT*LGSM[4, 3]SMLG[4, 1]BIN[30](a), SMLG[8, 3]LGSM[4, 0]BIN[30](a)}

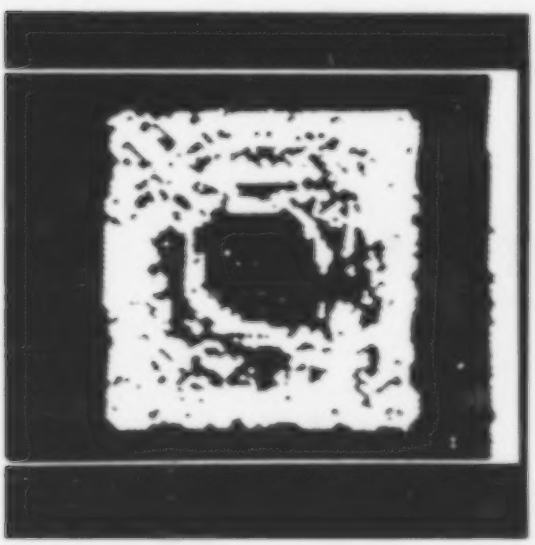


Figure 48. g = OR(NOT(d), f).

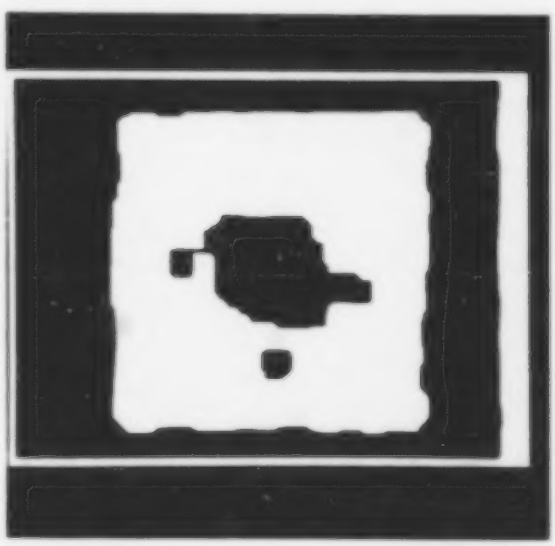


Figure 49. h = SMLG[8, 10]LGSM[8, 10](g).

Figure 50. The ashtray and test sheet with continuous gray levels on square black dot background.

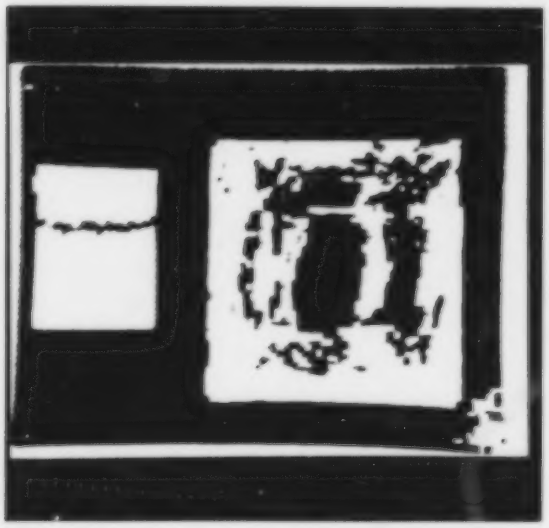


Figure 51. Result for square dot background.

In this case, the quality of the background sheet was not as good as in the case of Fig. 43, so there still remains much noise on the background in the result shown in Fig. 51.

The left part of Fig. 50 is a test sheet which has continuous gray levels. The result in Fig. 51 shows that we can detect the object even though the gray level changes from white to black; that is, we are using information other than the gray levels to detect the objects, which demonstrates another reason why this method is threshold insensitive. The threshold can be arbitrary if the objects have continuous and smooth gray level changes since the thresholded image is processed symmetrically. The preprocessing (P2-1) and (P2-2) was useful to make the resultant gap in the rectangular object smaller.

6.6 Discussion

As discussed in Section 2, the main reasons that a transparent object is seen are reflection, texture change by refraction, and shadow on the background from refraction. The reflection produces white regions (see the highlights in Fig. 43). In the case of a convex lens the refraction converts normal dots into larger spots (see the edge part of the ashtray in Fig. 43), and in the case of a concave lens, it converts normal dots into small black and white points (see the inside circular edge in Fig. 43). The small black and white points are converted into larger areas during

preprocessing by using a smaller factor in the erosion-dilatation type operation. The shadow created by the refraction has a tendency to make the image darker, hence the small points become black regions rather than white regions after the thresholding (see the inside edge of the ashtray in Fig. 43). Of course, if the transparent object is flat and does not change the texture, like the center portion of the ashtray, the local parallel operations cannot distinguish it from the background.

Table 6 shows the result for the black dot removal (P2), the preprocessing or erosion-dilatation (P2-1), and the main dilatation-erosion (P2-2). The main objective here is to convert black and white normal or unchanged dots into a white region. This is done by the main process. However, by this main process, black and white "point" areas, which happen typically when normal dots are reduced in size by the lens effect, are also converted into white regions. This is not wanted because the reduced points have their own information, that is, those are not the original dots. Here, the preprocess changes in advance the black and white point areas into black regions. The main process now does not change the black regions, and the "black" regions remain as information, while the normal dots are changed into "white" regions. Note here that in the process (P3) for the white dot removal, the preprocess (P3-1) changes the mixed points into "white," because the final objective there is to convert white dots into "black." The preprocess (P2-1) has another function, that is, it removes an isolated white noise "point." This makes the final result cleaner. Incidentally, an isolated black noise point in a white large area is automatically removed (changed to white) by the main process. Thus, the concatenation of the erosion-dilation operation and the dilatation-erosion operation becomes very reasonable when the dots which we want to remove have some known range of sizes.

In these experiments, the chessboard dots were first used because they have a symmetric property with respect to black and white shapes. This

Table 6. Results for the black dot removal

Before processing	Process	After processing
Black point	Main(P2-2)	White region
White point	Pre (P2-1)	Black region
Black dot	Main(P2-2)	White region
White dot		Almost unchanged
Black spot/region		Almost unchanged
White spot/region		Almost unchanged
Black/white points	Pre (P2-1)	Black region

symmetry makes LGSM-SMLG and SMLG-LGSM operations symmetric. If we use circular black dots, the white parts cannot be of the same shape, and the operations cannot be symmetric. The reason why the sheet was set diagonally is that the diamond shape is equi-distant in the four-neighbor sense and suitable for four-neighbor operations. Moreover, the equi-distant "circle" created by the four-neighbor operation has the smallest size among the eight-neighbor, octagonal (53), and Euclidian (57) distances. This means that the most precise structures can be detected by using the four-neighbor operations. We may be able to use different "regular" textures such as black dots (square or circular), white dots, black-and-white stripes, instead of the chessboard texture. However, in the stated sense, the diagonal chessboard arrangement may be the best.

The approach called "shape from texture" (58), utilizes the texture in a natural scene for extracting the shapes of objects. The method proposed here does not use the gray levels but rather an "artificial" texture for extracting the shapes. Hence we may call this method another kind of "shape from texture." Table 7 shows comparisons between the conventional shape from texture method and the proposed one. One example of the application of artificial texture is to project strips of light onto the scene. The strip image is segmented into regions by inverse Fourier transformation of selected frequency domains (59). In this case, we need the rather complex process of Fourier transformation, filtering, and inverse transformation, to analyze the projected texture. As another example, regular texture on an input object has been proposed (60). However, it is applicable to a more restricted environment than the proposed scheme because the processed object itself must have the regular property.

6.7 Conclusions

In this chapter, the conventional dilatation-erosion operations which are very suitable for commercial image processors were applied to detect transparent objects. By using a regularly dotted background, the sequential operations of dilation-erosion-erosion-dilatation and the symmetric

Table 7. Two kinds of "shape from texture"

Conventional shape from texture	Shape from texture in this chapter
Utilization of texture	Utilization of texture
Investigation of natural phenomenon and application to image processing	Application to image processing
Passive environment	Engineered environment
Scientific	Engineering

operations of erosion-dilatation-dilatation-erosion could be utilized and were unified to get more reliable results. It was shown that the method could be used not only for the transparent objects, but also for objects which have continuous gray levels. The method is directly applicable to scenes containing many isolated objects.

The authors proposed a recognition method for transparent objects by using dynamic programming techniques in the previous Chapters 3, 4, and 5. There, the model-based matching scheme was used to recognize the objects, and the original pattern itself was used as an input representation. Here, the results only indicate the regions of the objects from bottom-up processing. However, it is very helpful to use this kind of information as an input representation for the model-based matching. It supplies clearer input information to the matching stage and reduces the search space.

CHAPTER 7. CONCLUSIONS

In this report, Dynamic Programming was applied to 2D and 3D image recognition. Two methods were proposed. The inductive type DP described in Sections 3 and 4 was used for images of kidney structures photographed through a microscope, and for images of manufactured objects obtained by TV camera. The other is the rotation-free DP described in Section 5, which is used for the recognition of 3D objects. The main difference between these, as we have seen in Section 2, is the representation of the input image, i.e., pixel-wise representation in the former, and line-wise representation in the latter. The matching scheme cannot be discussed without specifying the representation since the matching is strongly related to how the features are extracted and how they are expressed. Here, the pixel-wise representation and the line-wise representation are the most popular and primitive but fundarmental ways of representing the contents of an image. In that sense, the proposed schemes covers a fairly wide range of applications of DP to two-dimensional images.

When DP is applied to a two-dimensional image, probably the biggest problem is how to make the DP method "real" in the 2D sense. We can say that there is no "real" 2D DP method in the same sense as the 1D DP is applied to 1D data. The next biggest problem is the number of calculations. Although the cost of calculations becomes cheaper and cheaper, still it is expensive for the 2D DP scheme due to the large number of pixels (or feature elements) on the 2D image and the even larger number of relationships between them. This is an inevitable consequence of the search for the global optimality. In Sections 3 and 4, the improvements were primarily concentrated on this aspect, i.e., to reduce the calculations by using a multi-resolutional framework.

The inductive DP method in Sections 3 and 4 and the rotation-free DP in Section 5 are contrasting in many aspects. One typical property is: The former permits occlusion but the latter does not and, of course, the latter is rotation-free while the former is not. These two properties are very important in the area of image recognition. The investigations of rotationally free methods in the framework of the inductive type or occlusion permitting DP in the context of line-wise representation may be one of the possible future studies.

The following improvements of the methods should also be done in the future: (1) Consideration of parallel property of the line in the model. This could be useful for both the white rings in the kidney pictures and for the brushes in the ashtry and ring pictures. (These were not included in the experiments.) (2) Consideration of area information along the line. In the case of 3D object recognition in Section 5, the area information is

stable and important. This information can be used in the DP process as one of the attributes of the line.

The methods proposed here fall into the rather "pure" category for matching problems in the pattern recognition problems. That is, the input representation is very simple, many alternatives were not investigated, and the process from feature extraction to matching is supposed to be straightforward. The iterative hypothesis-verification process was not assumed here. Of course, this kind of iteration improves the results to some extent, but it is only an improvement. When we want to have some large-scale step-up in the future, then fundamental changes must be made. In the case of iterative hypothesis-verificataion process, the strength of one hypothesis (i.e., matching) stage is important. If the matching method used in a stage is strong enough, then the verification can be simple even if we introduce the iterative stage, since the iteration can be small. If the matching method at each stage is not strong, then the number of hypothesis at each stage becomes very large and there exists the possibility of misjudgement at each stage. Consequently, the process becomes complex and the result becomes worse. Another merit of placing the problem in the simple environment is that the resultant problems with the method can be seen and analysed easily. This also helps in further investigation of the methods.

ACKNOWLEDGEMENTS

This work was done while one of the authors was visiting at the Division of Electrical Engineering of the National Research Council Canada, from the Electrotechnical Laboratory Japan, under the cooperative program between these organizations. The authors are most grateful to the persons concerned of both organizations for making this work possible. The authors are also indebted to Dr. Kazuhiko Yamamoto and Mr. Masahiko Matsuura for co-working on the original version of the method in Chapters 3 and 4, to Mrs. Cathy Merritt for co-working on Chapter 3, and to Dr. Michel Hospital for enthusiastic co-working on Chapter 5.

REFERENCES

1. R. Bellman. Dynamic Programming. Princetion University Press, New Jersey. 1957.

2. H. Ney. Dynamic programming as a technique for pattern recognition. Proc. 6th ICPR, Munich, Germany. 1982. pp. 1119-1125.

3. H. Yamada. Contour DP matching method and its application to hand-printed Chinese character recognition. Proc. 7th ICPR, Montreal, P.Q. 1984. pp. 389-392.

4. S. Fujiwara, K. Ikeda, Y. Tominaga, and T. Kiyono. On-line recognition of handwritten character by general contour vector sequence method. Inf. Process. (in Japan), 17(3): 191-199: 1976.

5. D. J. Burr. A technique for comparing curves. Proc. PRIP79, Chicago, IL. 1979. pp. 271-277.

6. K. Yoshida and H. Sakoe. Online handwritten character recognition by stack DP matching. Proc. Annual Conf. IECE of Japan, No.#90, 1983. pp. 1-90.

7. Y. Kondo, K. Abe, A. Fukumura. Decision method of similarity between directionally indexed sequence patterns. Proc. Annual Conf. IECE of Japan, p. 118, 1970.

8. Y. Fujimoto, S. Kadota, S. Hayashi, M. Yamamoto, S. Yajima, and M. Yasuda. Recognition of handprinted characters by nonlinear elastic matching", Proc. 3rd IJCPR, Coronado, California. 1976. pp. 113-118.

9. Y. Nakano, K. Nakata, Y. Uchikura, and A. Nakajima. Improvement of Chinese character recognition using projection profiles. Proc. 3rd IJCPR. Coronado, California. 1976. pp. 172-178.

10. Y. Isomichi and T. Ogawa. Pattern matching by using Dynamic Programming. Inf. Process. (in Japan), 16(1): 15-22: 1975.

11. H. Yamada, K. Yamamoto, and T. Saito. Normalization of directional pattern for directional histogram by the DP matching method. Proc. Annual Conf. IECE of Japan. No. 1571. 1985. p. 6-155.

12. V.A. Kovalevsky. Sequential optimization in pattern recognition and pattern description. Proc. IFIP Congr. 68. 1968. pp. 1603-1607.

13. H. Sakoe. Handwritten character recognition based on rubber string matching. Tech. Rep. IECE of Japan. No. PRL74-20. 1974.

14. H. Yamada and T. Kasvand. DP matching method for recognition of occluded, reflective and transparent objects with unconstrained background and illumination. Proc. 8th ICPR, Paris, France. October 1986. pp. 95-98.

15. F. Kimura, M. Yoshimura, and Y. Miyake. An algorithm for subpattern matching of line Patterms Proc. 6th ICPR, Munich, Germany. 1982. pp. 461-464.

16. M.A. Fischler and R.A. Elschlager. The representation and matching of pictorial structures. IEEE Trans. Comput., C-22(1): 67-92, 1973.

17. U. Montanari. On the optimal detection of curves in noisy pictures. Commun. ACM, 14(5): 335-345: 1971.

18. A. Martelli and U. Montanari. Optimal smoothing in picture processing: an application to fingerprints. Information Processing 71, North-Holland Publishing Co. 1972. pp. 173-178.

19. V. Velichko and N. Zagoruyko. Automatic recognition of 200 words. Int. J. Man-Machine Study, 2: 223-234; 1970.

20. H. Sakoe and S. Chiba. A dynamic programming approach to continuous speech recognition. Proc 7th Int. Congress Acoust., Budapest, Hungary. 1971. Paper 20C - 13.

21. H. Sakoe. Two-level DP matching - a dynamic programming-based pattern matching algorithm for connected word recognition. IEEE Trans. Acoust., Speech, Signal Process. ASSP-27(6): 558-595, 1979.

22. C.S. Myers and L.R. Rabiner. A level building dynamic time warping algorithm for connected word recognition. IEEE Trans. Acoust., Speech, Signal Process. ASSP-29(2): 284-297: 1981.

23. Y. Isomichi and T. Takemasa. Generation of hight map from the photo. Proc. 7th Image Eng. Conf. (in Japan), 1977. No. 3-1, pp. 31-34.

24. Y. Sasaki and T. Furukawa. Pairing the edges of stereograph by using dynamic programming technique. Tech. Rep. IECE of Japan. 1979. IE78-97.

25. Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. IEEE Trans. Pattern Anal. Mach. Intell.

PAMI-7(2): 139-154: 1985. Also Tech. Rep. CMU-CS-83-162, Carnegie-Mellon University, Oct. 1, 1983.

26. S.A. Lloyd. Waveform matching using multiresolution dynamic programming. Pattern Recognition Lett. 2: 1281-1283: 1984.

27. H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech, Signal Process. ASSP-26(1): 43-49; 1978.

28. Y. Kitahara and Y. Isomichi. Online handprinted Kanji recognition using directional cosine DP matching, Trans. IECE of Japan. 1981. Vol. J64-D, pp. 962-969.

29. C.C. Tappert. Cursive script recognition by elastic matching. IBM J. Res. Dev. 26, (6): 765-771; 1982.

30. T. Wakahara and M. Umeda. On-line character recognition using selective stroke connection, Proc. Annual Conf. IECE Japan, No.S10-8, p. 5-413.

31. P-Y. Lu and R.W. Brodersen. Real-time on-line symbol recognition using a DTN processor. Proc. 7th ICPR, Montreal, P.Q. 1984. pp. 1281-1283.

32. H. Decker. Automatic positioning of objects using edges. Proc. 6th ICPR, 1982. pp. 850-852.

33. U. Montanari, A note on minimal length polygonal approximation to a digitized contour. Commun. ACM, 13(1): 41-47; 1970.

34. D.H. Ballard and J. Sklansky. A ladder-structured decision tree for recognizing tumors in chest radiographs. IEEE Trans. Comput. C-25(5): 503-513; 1976.

35. Y. Sato, H. Kawasaki, and H. Obata. Contour extraction method from dynamic images by using dynamic programming. Proc. 12th Image Eng. Conf. (in Japan), 1981. pp. 17-20,

36. H. Yamada, K. Yamamoto, and M. Matsuura. Recognition of ultrasonic human kidney organ images by the two-dimensional DP matching method. Trans. IECE of Japan. 1985. J68-D, No. 10. pp. 1649-1656.

37. D.H. Hubel and T.N. Wiesel. Functional architecture of macaque monkey visual cortex. Proc. R. Soc. London, Ser. B, 198: 1-59, 1977.

38. M. Oshima and Y. Shirai. Object recognition using three-dimensional information. IEEE Trans. Pattern Anal. Machine Intell. PAMI-5(4): 353-361, 1983.

39. W.A. Perkins. A model-based vision system for industrial parts. IEEE Trans. Comput. C-27(2): 126-143; 1978.

40. N. Ayashe and O.D. Faugeras. A new method for the recognition and positioning of 2-D objects. Proc. 7th Int. Conf. Pattern Recog., Montreal, P.Q. 1984. pp. 1274-1277.

41. J.L. Turney, T.N. Mudge, and R.A. Volz. Recognizing partially occluded parts. IEEE Trans. Pattern Anal. Machine Intell. PAMI-7(4): 410-421; 1985.

42. T. Kasvand, R. Oka, M. Dhome and M. Rioux. Segmentation, learning, and recognition of objects in range camera images. 4th Canadian CAD/CAM and Robotics Conference, June, 1985.

43. H. Yamada, M. Hospital, and T. Kasvand. Rotation-invariant contour DP matching method for 3D object recognition. Proc. Int. Conf. on Systems, Man and Cybernetics, Atlanta, GA. October, 1986. pp. 997-1001.

44. M. Hospital and M. Richetin. Segmentation en bandes d'images grises par segmentation polygonale des contours binaires orientes. Proc. 5th AFCET Pattern Recognition and Artificial Intelligence Conference, Grenoble, France Nov. 1985. pp. 179-186.

45. M. Hospital and M. Richetin. Characterization of some drilled pit images from contour and ridge point detection. Proc. CVPR86, Miami, FL. June 1986.

46. M. Richetin, P. Saint-Marc, and J.T. Lapereste. Describing greylevel textures through curvature primal sketching. Proc. ICASSP, Tokyo, 1986. pp. 1433-1436.

47. F. Metelli. The perception of transparency. Sci. Am. 230(4): 90-98; 1974.

48. D.S. Kay and D. Greenberg. Transparency for computer synthesized images. Computer Graphics, Vol.13, No.2, pp. 158-164, 1979.

49. R.J. Woodham. Photometric stereo: a reflectance map technique for determining surface orientation from image intensity. Proc. SPIE 22nd Technical Symposium, San Diego, CA. 1978. pp. 155, 136-143.

50. R.B. Kelley, J.R. Birk, H.A.S. Martins, and R. Tella. A robot system which acquires cylindrical workpieces from bins. IEEE Trans. Syst. Man. Cybern. SMC-12(2): 204-213; 1982.

51. M. Ando, K. Mita, and T. Inagaki. Automatic optical through hole inspection method for printed wiring boards using leakage light detection. IEEE Int. Conf. Robotics and Automation. St. Louis, MO. 1985. pp. 481-487.

52. J. Beck. Perception of transparency in man and machine computer vision. Comput. Vision, Graphics, Image Process. 31: 127-138; 1985.

53. A. Rosenfeld and J.L. Pfalts. Distance functions on digital pictures. Pattern Recognition, 1: 33-61; 1968.

54. M. Ejiri, T. Uno, M. Mese, and S. Ikeda. A process for detecting defects in complicated patterns. Computer Graphics Image Process. 2: 326-339; 1973.

55. M.J.B. Duff. Special hardware for pattern recognition. Proc. 6th ICPR, Munich, Germany, 1982. pp. 368-379.

56. S.R. Sternberg. Parallel architectures for image processing. Real Time/Parallel Computing: Image Processing. Edited by K. Preston. Plenum Press, New York. 1980.

57. H. Yamada. Complete Euclidean distance transformation by parallel operation. Proc. 7th ICPR. 1984, Montreal, P.Q. 1984. pp. 69-71.

58. J.J. Gibson. The Perception of the Visual World. Houghton Mifflin, Boston, MA. 1950.

59. P.M. Will and K.S. Pennington. Grid coding: a preprocessing technique for robot and machine vision. Proc. 2nd IJCAI, London. 1971. pp. 66-70.

60. K. Ikeuchi. Shape from regular patterns (an example of constaint propagation in vision). Proc. 5th IJCPR. Miami, FL. 1980. pp. 1032-1039.

APPENDIX A. GMAT - INDUCTIVE DYNAMIC PROGRAMMING MATCHING PROGRAM

A.1. Introduction

GMAT is a matching program for gray level images by using the inductive type DP matching algorithm, written in FORTRAN on VAX11/780 under the user id [YAMADA]. This was used in Chapters 3 and 4. This program uses two types of files: one is an input image file, and the other is a work file. The input image is taken from the TV camera by the PICACQS program in advance. The format of the image is 640 x 480, 8bits/pixel, and 1 record = 1 row. The work file is used to store parameters, shape models, and matching results. Each record in the work file has fixed length. Shape model is created by GMAT command "S" interactively, and stored as one record in the work file in advance. Multiple input images or different work files can be specified by different file numbers. The GMAT program also uses the Norp ak RGP3050 display as its standard environment. The flow of the program is controled by interactive command and parameters. The parameter(s) can be changed by the command I(integer), F(floating), or R(read from work file). The command is explained in Section A.2, and the parameter is explained in Table A1 and in Section A.3 together with some examples.

A.2. Interactive Command

When the program asks by "Key?", the operator must reply by one digid alphabet. The alphabet must be upper-case!

(1) Preparation (file assignment) before GMAT program is run. Usually done by the login procedure in the file LOGIN.COM.

```
$ ASS [YAMADA.D]KIDG1.DAT FOR031   !Kidney with large magnification
factor.
$ ASS [YAMADA.D]KIDG2.DAT FOR032   !Kidney with small magnification
factor.
$ ASS [YAMADA.D]DOTS1.DAT FOR040   !
$ ASS [YAMADA.D]SCIS1.DAT FOR041   !Scisors and matal rings (learning).
$ ASS [YAMADA.D]SCIS2.DAT FOR042   !Scisors and metal rings (test-1).
$ ASS [YAMADA.D]SCIS3.DAT FOR043   !Scisors and metal rings (test-2).
$ ASS [YAMADA.D]SCIS4.DAT FOR044   !Scisors and metal rings (test-3).
$ ASS [YAMADA.D]ASHT1.DAT FOR046   !Ashtrays (for learning).
$ ASS [YAMADA.D]ASHT2.DAT FOR047   !Ashtrays with black background.
$ ASS [YAMADA.D]ASHT3.DAT FOR048   !Ashtrays with magazine background.
$ ASS [YAMADA.D]ASHT4.DAT FOR049   !Ashtrays with floor background.
$ ASS [YAMADA.D]BDOT1.DAT FOR051   !Ashtray with square dot background.
$ ASS [YAMADA.D]WDOT1.DAT FOR052   !Ashtray with white dot background.
```

```
$ ASS [YAMADA.D]BCKBL.DAT FOR053   !Asctray with black background.
$ ASS [YAMADA.D]BCKGR.DAT FOR054   !Ashtray with gray backbround.
$ ASS [YAMADA.D]BCKWH.DAT FOR055   !Ashtray with white background.
$ ASS [YAMADA.D]POLAR.DAT FOR056   !Ashtray with polarized lighting from
top.
$ ASS [YAMADA.D]DIFBT.DAT FOR057   !Ashtray with diffused lighting from
bottom.
$ ASS [YAMADA.D]CHCK1.DAT FOR058   !Ashtray with checkboard background-1.
$ ASS [YAMADA.D]CHCK2.DAT FOR059   !Ashtray with checkboard background-2.
$ ASS [YAMADA.D]RSCI1.DAT FOR060   !Work file for Scissors and metals-1.
$ ASS [YAMADA.D]RSCI2.DAT FOR062   !Work file for Scissors and metals-2.
$ ASS [YAMADA.D]RASH1.DAT FOR063   !Work file for Ashtrays-1.
$ ASS [YAMADA.D]RASH2.DAT FOR064   !Work file for Ashtrays-2.
$ ASS [YAMADA.D]RKID1.DAT FOR061   !Work file for Kidneys.
```

(2) "GMAT" job start, fundamental operations and stop. Here, (cr) means carrage return. The line which has (cr) is input by the operator. Statement after ! mark is only a comment for the explanation. Other lines are outputs from the computer.

```
 $ RUN GMAT(cr)
KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/63)
 I(cr)     !Integer parameter set.
 1,63(cr)  !First parameter is set to 63 (this specifies work file number).
 ><FL(           1)OLD=          60 NEW=          63
 R(cr)  !Read parameters from work file number 63.
 9(cr)  !Record number is 9.
KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/63)
  D(cr)  !Display control. This must be done before the display is used.
  INIT 1=CLEAR, 0=NO-CLEAR, 2=SKIP, -1=END
 1(cr)  !2=Go to next (Set bank), -1=Go to Key? mode.
  SET BANK 1, 2, 3, OR 4=SKIP, 0=END
 2(cr)
  OVLY 1, 2, 3, 4, (-N=DISABLE), OR 5=SKIP, -5=END
 1(cr)
  COLOR 0,1,2,...,7 OR 8=SKIP, -1=END
 7(cr)
  CLEAR OVLY 1(=Y), 0(=N)
 1(cr)
 KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/63)
 P(cr)  !Print parameters
 ><FL 1=     63 ><RS 2=      0 >IFL 3=     46 >RES 4=      0 >MSK 5= 0
 ><IN 6=      0 MAXS31=      0 MAXV32=    0.0
 HGEQ11=     42 IREC12=      1 PROC13=      0 PICH14=      8 !INP15=      1
 ETH116=    0.0 ETH217=    1.0 ISIZ18=      0 CONF19=    0.0 DIFF20=    0.0
 INX 21=    640 INY 22=    480 I1X 23=      1 I1Y 24=      1 I3X 25=    640
```

```
I3Y 26=    480 JNX 27=     80 JNY 28=     60 IXSB29=      0 IYSB30=     0
LMRE42=      1 LANG43=      1 LMAX44=      1 MSW 47=       0     48=     0
!RES49=      0 FREZ50=      4 ROU 51=   28.0 EPSL52=    999 ROTA53=     0
WRKF61=     29 MPRE62=      8 LAMD63=   2.0 OMEG64=      2 INCL65=   0.1
GAMA66=    1.0 DUMP67=      5 DMPX68=     14 DMPY69=     10 OMG270=     0
DELT71=    0.5 MDFY72=   -0.5 MOD 73=    1.0 DTH174=    0.0 DTH275=   0.5
MNS2 1=     16 MID2 2=      0 LEN2 3=    0.0 XC 2 4=    0.0 YC 2 5=   0.0
RESULT=     0     0     0     0     0     0    MNS=    16    NO=     0
KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/63)
G(cr)  !Go. Execute program by using the current parameters.
       !Fig. 25 is displayed by the parameters in this case.
 JXMN,JYMN,JXMX,JYMX            0           0         641         481
 WARNING. REF. NOT SET
NO=  0 SMLRTY=   0 .99SEC=      0 END=     0     0 BGN=     0     0 FIL=63
  1=   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
  9=   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/63)
R(cr)  !Read parameters from work file number 63.
1(cr)  !Record=1 (Model shape has been stored in record 1).
 KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/63)
M(cr)  !Model contents list.
RESULT=     0     0     0     0     0     0    MNS=    16    NO=     0
MASK-HEAD   MNS= 16 ID=   0 LEN=  376.99 XY=  376.00  140.39
  1   23.1   4.6  23.6   0.2 399.1  84.6      0     0     0     0     0
  2   19.6  13.1  23.6   0.6 418.7  97.7      0     0     0     0     0
  3   13.1  19.6  23.6   1.0 431.8 117.3      0     0     0     0     0
  4    4.6  23.1  23.6   1.4 436.4 140.4      0     0     0     0     0
  5   -4.6  23.1  23.6   1.8 431.8 163.5      0     0     0     0     0
  6  -13.1  19.6  23.6   2.2 418.7 183.1      0     0     0     0     0
  7  -19.6  13.1  23.6   2.6 399.1 196.2      0     0     0     0     0
  8  -23.1   4.6  23.6   2.9 376.0 200.8      0     0     0     0     0
  9  -23.1  -4.6  23.6   3.3 352.9 196.2      0     0     0     0     0
 10  -19.6 -13.1  23.6   3.7 333.3 183.1      0     0     0     0     0
 11  -13.1 -19.6  23.6   4.1 320.2 163.5      0     0     0     0     0
 12   -4.6 -23.1  23.6   4.5 315.6 140.4      0     0     0     0     0
 13    4.6 -23.1  23.6   4.9 320.2 117.3      0     0     0     0     0
 14   13.1 -19.6  23.6   5.3 333.3  97.7      0     0     0     0     0
 15   19.6 -13.1  23.6   5.7 352.9  84.6      0     0     0     0     0
 16   23.1  -4.6  23.6   6.1 376.0  80.0      0     0     0     0     0
KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/63)
E(cr)  !End of program.
FORTRAN STOP
$
```

(3) Other commands. These are also available at "Key?" mode.

```
W(cr)  !Write parameters to work file, similar to R command.
F(cr)  !Floating parameter set, similar to I command.
S(cr)  !Set model interactively.
```

A.3. Example of Parameters

The function of each parameter is explained in Table A1.

Table A1. Parameters in GMAT.

Name	No.	Example	Contents
f<FL	1=	64	Work file number
><RS	2=	111	Result output record on work file
>IFL	3=	47	Input image file number
>RES	4=	0	Result input record (file number = parameter 1)
>MSK	5=	2	Model input record (file number = parameter 1)
><IN	6=	0	
HGEQ	11=	1	If 1, then do histogram equalization
IREC	12=	1	
PROC	13=	8	
PICH	14=	8	Sampling pitch
!INP	15=	0	Input pattern display 1 = Gray level pattern, 2 = Sampled pattern, 3 = Edge amplitude, 0 = No-display.
ETH1	16=	0.0	Lower threshold for edge amplitude
ETH2	17=	1.0	Higher threshold for edge amplitude
ISIZ	18=	0	
CONF	19=	0.0	Parameter for Huckel edge operator
DIFF	20=	0.0	Parameter for Huckel edge operator
INX	21=	640	Input pattern size x
INY	22=	480	Input pattern size y
I1X	23=	1	Initial x position for sampling
I1Y	24=	1	Initial y position for sampling
I3X	25=	640	X length for sampling
I3Y	26=	480	Y length for sampling
JNX	27=	80	Size x of sampled pattern
JNY	28=	60	Size y of sampled pattern
IXSB	29=	320	
IYSB	30=	280	
MAXS	31=	61253	Maximum of sampled pattern (Computer output)
MAXV	32=	531.3	Maximum of edge amplitude (Computer output)
LMRE	42=	1	Number of parameter files executed in the job
LANG	43=	9	Number of different angle models
LMAX	44=	2	Number of extracting results (maxima)
MSW	47=	2	Matching process control

```
                        1=Gray level only, 2=Edge only,
                        3=Both gray level and edge.
      48=      43
!RES  49=       0   Width of displayed result
FREZ  50=       4   Blocking width
ROU   51=     0.0
EPSL  52=     999
ROTA  53=       0   Rotational increment when models are generated
WRKF  61=      29
MPRE  62=       8
LAMD  63=     1.2   Variation of length in matching (λ)
OMEG  64=       2   Variation of width in matching (ω)
INCL  65=     0.1
GAMA  66=     1.0   Constant for gap in the matching
DUMP  67=       1   Dump control
DMPX  68=      14
DMPY  69=      10
OMG2  70=       0
DELT  71=     0.5
MDFY  72=    -0.5
MOD   73=     1.0   Modulo for angle calculation. 1.0 means pai.
DTH1  74=     0.1
DTH2  75=     0.3
MNS  201=      16   Number of model segments
MID  202=       0
LEN  203=   400.0
XC   204=   428.0
YC   205=   256.0
```

---

(1) Original pattern display. Input pattern file = 31. Color display must be initialized in advance by using "D" command. Figure 14 (without results) is displayed by the G command after setting these parameters. This parameter set is existing in file=61, record=9.

```
><FL 1=      61 ><RS 2=       0 >IFL 3=      31 >RES 4=       0 >MSK 5=
><IN 6=       0 MAXS31=       0 MAXV32=     0.0
 HGEQ11=     42 IREC12=       1 PROC13=       0 PICH14=       8 !INP15=
 ETH116=    0.0 ETH217=     1.0 ISIZ18=       0 CONF19=     0.0 DIFF20=     0
 INX 21=    640 INY 22=     480 I1X 23=       1 I1Y 24=       1 I3X 25=     6
 I3Y 26=    480 JNX 27=      80 JNY 28=      60 IXSB29=       0 IYSB30=
 LMRE42=      1 LANG43=       1 LMAX44=       1 MSW 47=       0      48=
 !RES49=      0 FREZ50=       4 ROU 51=    28.0 EPSL52=     999 ROTA53=
 WRKF61=     29 MPRE62=       8 LAMD63=     2.0 OMEG64=       2 INCL65=    0.1
 GAMA66=    1.0 DUMP67=       5 DMPX68=      14 DMPY69=      10 OMG270=      0
 DELT71=    0.5 MDFY72=    0.5 MOD 73=     1.0 DTH174=     0.0 DTH275=    0.5
```

```
MNS2 1=     16 MID2 2=      0 LEN2 3=    0.0 XC 2 4=    0.0 YC 2 5=    0.0
RESULT=     0     0     0     0     0     0    MNS=    16    NO=     0
```

(2) Sampled pattern display. Input pattern file(3) = 33, sampling pitch(14) = 16. Figure 8 is displayed by this parameter set. This parameter set is existing in file 61, record 8.

```
><FL 1=     61 ><RS 2=      0 >IFL 3=     33 >RES 4=      0 >MSK 5=     0
><IN 6=      0 MAXS31=  53228 MAXV32=    0.0
HGEQ11=     42 IREC12=      1 PROC13=     1 PICH14=     16 !INP15=     2
ETH116=    0.0 ETH217=    1.0 ISIZ18=     0 CONF19=    0.0 DIFF20=   0.0
INX 21=    640 INY 22=    480 I1X 23=     1 I1Y 24=      1 I3X 25=    640
I3Y 26=    480 JNX 27=     40 JNY 28=     30 IXSB29=     0 IYSB30=     0
LMRE42=      1 LANG43=     1 LMAX44=     1 MSW 47=      0     48=     0
!RES49=      0 FREZ50=     4 ROU 51=  28.0 EPSL52=    999 ROTA53=     0
WRKF61=     29 MPRE62=     8 LAMD63=   2.0 OMEG64=      2 INCL65=   0.1
GAMA66=    1.0 DUMP67=     5 DMPX68=    14 DMPY69=     10 OMG270=     0
DELT71=    0.5 MDFY72=   0.5 MOD 73=   1.0 DTH174=    0.0 DTH275=   0.5
MNS2 1=     16 MID2 2=      0 LEN2 3=    0.0 XC 2 4=    0.0 YC 2 5=    0.0
RESULT=     0     0     0     0     0     0    MNS=    16    NO=     0
```

(3) Execution, kidney, scalar, first-stage, sampling-pitch(14)=16. Model: file(1)=61, record(5)=1(Large circle model). (Record-2 = Small circle model). By the command G after setting this current parameters, the DP matching is done. The resultant shapes together with the current parameters are written in file number(1)=61 and record numbers(2) from 11 to 15 because 5 maxima is specifiled by parameter 44. This record can be used as the shape model for the following DP matching under the constraint of this result.

```
><FL 1=     61 ><RS 2=     11 >IFL 3=     33 >RES 4=      0 >MSK 5=     1
><IN 6=      0 MAXS31=  53228 MAXV32=    0.0  HGEQ11=     33 IREC12=
1 PROC13=      1 PICH14=     16 !INP15=     0
ETH116=    0.0 ETH217=    1.0 ISIZ18=     0 CONF19=    0.0 DIFF20=   0.0
INX 21=    640 INY 22=    480 I1X 23=     1 I1Y 24=      1 I3X 25=    640
I3Y 26=    480 JNX 27=     40 JNY 28=     30 IXSB29=     0 IYSB30=     0
LMRE42=      1 LANG43=     1 LMAX44=     5 MSW 47=      1     48=    11
!RES49=      0 FREZ50=     4 ROU 51=   0.0 EPSL52=    999 ROTA53=     0
WRKF61=     29 MPRE62=     8 LAMD63=   2.0 OMEG64=      1 INCL65=   0.1
GAMA66=    1.0 DUMP67=    21 DMPX68=    14 DMPY69=     10 OMG270=     0
DELT71=    0.5 MDFY72=   0.5 MOD 73=   1.0 DTH174=    0.0 DTH275=   0.5
MNS2 1=     16 MID2 2=      0 LEN2 3=  556.0 XC 2 4=  504.0 YC 2 5= 377.0
RESULT=   881  2759    31    18    31    18    MNS=    16    NO=    11
```

(4) Result shape display. Output width(49) = 1. This parameter set display only the resultant contour which exists in the file number(1)=61 and record number(4)=11. The original image must be displayed by using another set of parameters like the example (1). The image written by this parameter set on the image of parameter set (2) makes complete Fig. 8.

```
><FL 1=     61 ><RS 2=      0 >IFL 3=      0 >RES 4=     11 >MSK 5=      0
><IN 6=      0 MAXS31=      0 MAXV32=    0.0
HGEQ11=     42 IREC12=      0 PROC13=      0 PICH14=      1 !INP15=      0
ETH116=    0.0 ETH217=    1.0 ISIZ18=      0 CONF19=    0.0 DIFF20=    0.0
INX 21=    640 INY 22=    480 I1X 23=      1 I1Y 24=      1 I3X 25=    640
I3Y 26=    480 JNX 27=    640 JNY 28=    480 IXSB29=      0 IYSB30=      0
LMRE42=      1 LANG43=      1 LMAX44=      1 MSW 47=      0     48=      0
!RES49=      1 FREZ50=      4 ROU 51=   28.0 EPSL52=    999 ROTA53=      0
WRKF61=     29 MPRE62=      8 LAMD63=    2.0 OMEG64=      2 INCL65=    0.1
GAMA66=    1.0 DUMP67=     21 DMPX68=     14 DMPY69=     10 OMG270=      0
DELT71=    0.5 MDFY72=   -0.5 MOD 73=    1.0 DTH174=    0.0 DTH275=    0.5
MNS2 1=     16 MID2 2=      0 LEN2 3=  556.0 XC 2 4=  504.0 YC 2 5=  377.0
RESULT=   881  2759    31    18    31    18    MNS=    16    NO=     0
```

(5) Execution, kidney, second stage, sampling pitch(14) = 4. Input pattern file(3)=33. Former result : file(1) = 61, record(5) = from 11 to 15. Result output : record(2) = 31 to 35. The first result (record = 31) obtained by this parameter set is shown in Fig. 9.

```
><FL 1=     61 ><RS 2=     31 >IFL 3=     33 >RES 4=      0 >MSK 5=     11
><IN 6=      0 MAXS31=   3657 MAXV32=   84.1
HGEQ11=     33 IREC12=      1 PROC13=      1 PICH14=      4 !INP15=      0
ETH116=    0.0 ETH217=    1.0 ISIZ18=      0 CONF19=    0.0 DIFF20=    0.0
INX 21=    640 INY 22=    480 I1X 23=      1 I1Y 24=      1 I3X 25=    640
I3Y 26=    480 JNX 27=    160 JNY 28=    120 IXSB29=      0 IYSB30=      0
LMRE42=      5 LANG43=      1 LMAX44=      1 MSW 47=      1     48=     21
!RES49=      0 FREZ50=      4 ROU 51=    0.0 EPSL52=      4 ROTA53=      0
WRKF61=     29 MPRE62=      8 LAMD63=    1.2 OMEG64=      1 INCL65=    0.1
GAMA66=    1.0 DUMP67=      1 DMPX68=     14 DMPY69=     10 OMG270=      0
DELT71=    0.5 MDFY72=    0.5 MOD 73=    2.0 DTH174=    0.3 DTH275=    0.5
MNS2 1=     16 MID2 2=      0 LEN2 3=  566.2 XC 2 4=  500.5 YC 2 5=  377.5
RESULT=   878  1439   122    69   122    69    MNS=    16    NO=    31
```

(6) Execution, another kidney, first stage. Sampling pitch(14) = 8. Input pattern file(3) = 31. Result output : record(2) = 81 to 90. The results obtained by this parameter set are shown in Fig. 3-8.

```
>(FL 1=     61 >(RS 2=     81 >IFL 3=     31 >RES 4=      0 >MSK 5=      2
>(IN 6=      0 MAXS31=   9741 MAXV32=   84.1
HGEQ11=     31 IREC12=      1 PROC13=      4 PICH14=      8 !INP15=      0
ETH116=    0.0 ETH217=    1.0 ISIZ18=      0 CONF19=    0.0 DIFF20=    0.0
INX 21=    640 INY 22=    480 I1X 23=      1 I1Y 24=      1 I3X 25=    640
I3Y 26=    480 JNX 27=     80 JNY 28=     60 IXSB29=      0 IYSB30=      0
LMRE42=      1 LANG43=      1 LMAX44=     10 MSW 47=      3      48=     21
!RES49=      0 FREZ50=      4 ROU 51=    0.0 EPSL52=    999 ROTA53=      0
WRKF61=     29 MPRE62=      8 LAMD63=    2.0 OMEG64=      1 INCL65=    0.1
GAMA66=    1.0 DUMP67=     21 DMPX68=     14 DMPY69=     10 OMG270=      0
DELT71=    1.0 MDFY72=    0.5 MOD 73=    2.0 DTH174=    0.0 DTH275=    1.0
MNS2 1=     16 MID2 2=      0 LEN2 3=  240.4 XC 2 4=  560.0 YC 2 5=  270.5
RESULT=  881 23750    70    29    70    29    MNS=    16    NO=    81
```

(7) Execution. Input pattern file = 43(scisors and metals). Model: file(1)=61, record(5)=1(Thumb side of scisors). The resultant shapes file number(1) = 61, record number(2) = from 91 to 126. Thirty-six different rotational model is generated automatically because 36 is specified in parameter 43. These 36 results are sorted by the resultant evaluation value and the maximum is taken as the first result. The first result of the thumb-side is shown in Fig. 24.

```
>(FL 1=     60 >(RS 2=     91 >IFL 3=     43 >RES 4=      0 >MSK 5=      6
>(IN 6=      0 MAXS31=  63260 MAXV32=  316.9
HGEQ11=     43 IREC12=      1 PROC13=      8 PICH14=      8 !INP15=      0
ETH116=    0.0 ETH217=    1.0 ISIZ18=      0 CONF19=    0.0 DIFF20=    0.0
INX 21=    640 INY 22=    480 I1X 23=      1 I1Y 24=      1 I3X 25=    640
I3Y 26=    480 JNX 27=     80 JNY 28=     60 IXSB29=    320 IYSB30=    240
LMRE42=      1 LANG43=     36 LMAX44=      1 MSW 47=      2      48=     46
!RES49=      0 FREZ50=      4 ROU 51=    0.0 EPSL52=    999 ROTA53=      0 t
WRKF61=     29 MPRE62=      8 LAMD63=    1.2 OMEG64=      1 INCL65=    0.1
GAMA66=    1.0 DUMP67=      1 DMPX68=     14 DMPY69=     10 OMG270=      0
DELT71=    0.5 MDFY72=   -0.5 MOD 73=    1.0 DTH174=    0.1 DTH275=    0.3
MNS2 1=     16 MID2 2=      0 LEN2 3=  890.9 XC 2 4=  208.5 YC 2 5=  249.0
RESULT=  728 36672    32     4    32     4    MNS=    16    NO=    91
```

(8) Execution. Input pattern file = 47(Ashtrays). Histogram equalization. Model: file(1)=64, record(5)=2(Outer square). The resultant shapes: file number(1) = 61, record number(2) = from 111 to 128. Nine different rotational model is generated automatically because of the parameter(43)=9, and the first and second maxima of each rotational angle is stored because of the parameter(44)=2. These 18 results are sorted by the resultant evaluation value and the first several results are shown in Fig. 29.

```
 ><FL 1=     64 ><RS 2=    111 >IFL 3=     47 >RES 4=      0 >MSK 5=     2
 ><IN 6=      0 MAXS31=  61253 MAXV32=  531.3  HGEQ11=      1 IREC12=    1
1 PROC13=     8 PICH14=      8 !INP15=     0
 ETH116=    0.0 ETH217=    1.0 ISIZ18=     0 CONF19=    0.0 DIFF20=   0.0
 INX 21=    640 INY 22=    480 I1X 23=     1 I1Y 24=      1 I3X 25=   640
 I3Y 26=    480 JNX 27=     80 JNY 28=     60 IXSB29=   320 IYSB30=   280
 LMRE42=      1 LANG43=     9 LMAX44=     2 MSW 47=      2      48=    43
 !RES49=      0 FREZ50=     4 ROU 51=   0.0 EPSL52=    999 ROTA53=     0
 WRKF61=     29 MPRE62=     8 LAMD63=   1.2 OMEG64=      2 INCL65=   0.1
 GAMA66=    1.0 DUMP67=     1 DMPX68=    14 DMPY69=     10 OMG270=     0
 DELT71=    0.5 MDFY72=  -0.5 MOD 73=   1.0 DTH174=    0.1 DTH275=   0.3
 MNS2 1=     16 MID2 2=     0 LEN2 3=  400.0 XC 2 4=  428.0 YC 2 5= 256.0
 RESULT=  893 39583    48     26     48     26     MNS=    16     NO=   111
KEY?(END,R-PAR,W-PAR,ISET,FSET,PLST,MLST,ST-REF,DISP,GO,(I.1,60/61/62/6 3)
```

A.4. Utilities and files

(1) Compile and link.

```
$ FORT GMAT
$ LIN GMAT,[YAMADA.L]USRLIB/LIB,-
      [YAMADA.L]CREID/LIB,[SAKANE]SPIDER/LIB,-
      [MERRITT.SUBLIB]HOUSE/LIB,[BINCH.RGP3050.VAX]ZLIB3050/LIB,-
      [YAMADA.L]PLOT10/LIB
$
```

(2) VAX user files under DRB1:[YAMADA]

```
[YAMADA]B.DIR;1              !For batch job submit
[YAMADA]C.DIR;1              !Command files
[YAMADA]D.DIR;1              !Data of images
[YAMADA]ETL.DIR;1            !ETL programs
[YAMADA]L.DIR;1              !Load modules and Library
[YAMADA]M.DIR;1              !Main programs
[YAMADA]S.DIR;1              !Sub-programs
[YAMADA]T.DIR;1              !Text data
[YAMADA]VDP.DIR;1            !For plot-out to display
[YAMADA]BDOT.EXE;1           !Parallel operation on regular background.
[YAMADA.C]UALLOC.COM;38      !For file allocation
[YAMADA.C]ULIB.COM;1         !Update subroutine library
[YAMADA.C]ULINK.COM;2        !Main program link
```